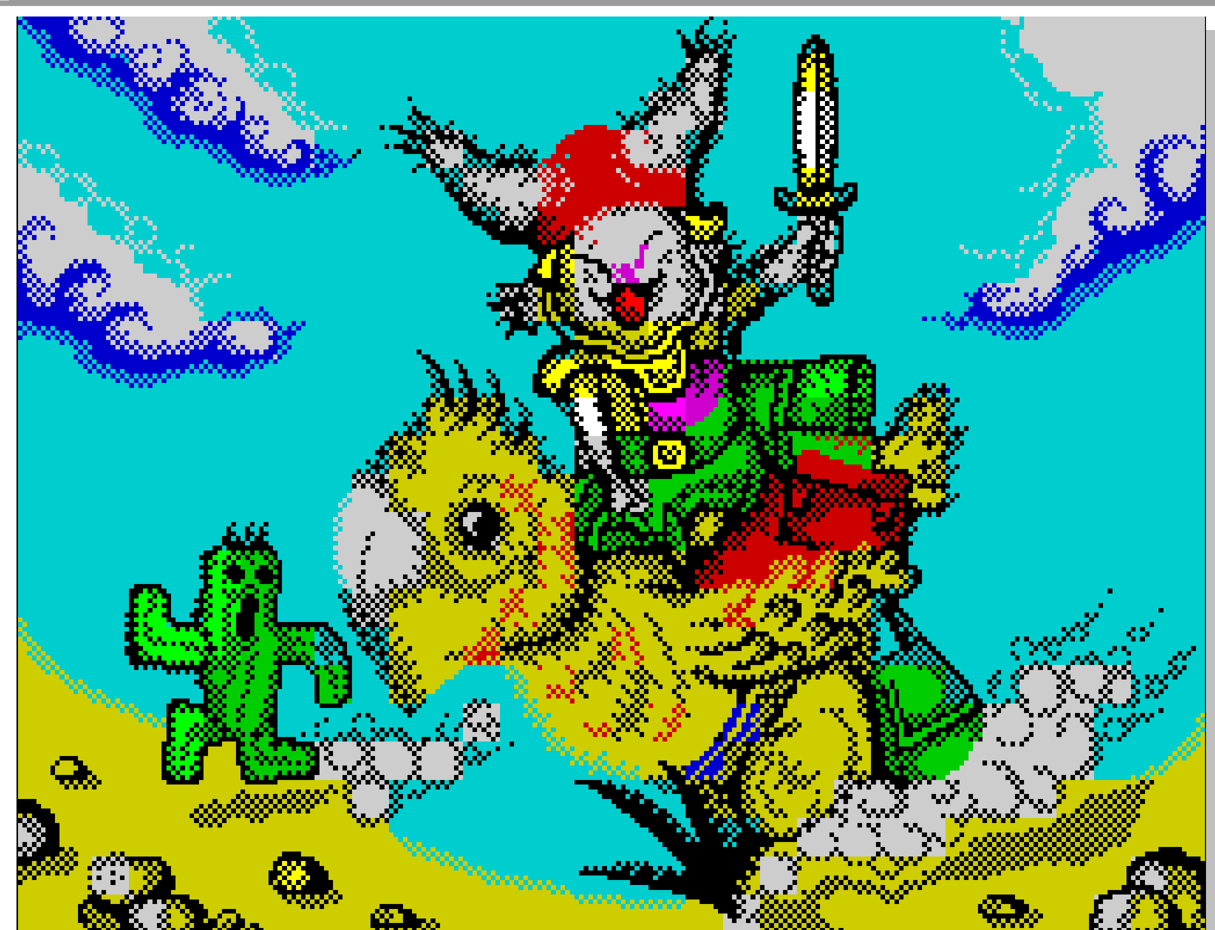
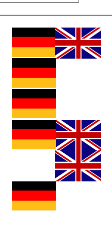




Angriff auf die nächsten 30 Jahre!



Herausgeber und für den Inhalt verantwortlicher:
Leszek Chmielewski, Prager Straße 92/11/12, 1210 Wien, Österreich
@Mail: retrozx@gmail.com
Klubkonto (Inhaber: Bernhard Lutz):IBAN: DE59 5486 2500 0000 5461 43
SWIFT-Code: GENODE6K, BIC-Code: GENODE61SUW
KTO.: 546143, BLZ: 54862500 (VR Bank Südpfalz, Sitz: Landau)

Ausgabe 229

2 Quartal 2012

Das Vorwort

http://www.womoteam.de/
http://spc.tlienhard.com/

Willkommen zu der Zeitschrift von Usern für User. Wir sind vor allem auf EURE Artikel angewiesen. Ich kann alleine keine (angepeilten) 24-32 Seiten füllen, so gerne ich es auch tun würde. Ehrenwort! Für eingeschickte Artikel gelten folgende Regeln:
Die Artikel müssen sich mit dem Spectrum, ZX81, SAM Coupé, Sprinter 2000 oder nahen Verwandten des Sinclair ZX Spectrum befassen, auch Artikel über passende Hardware und Software sind gerne gesehen.
MAC/PC Software: Nur wenn ausdrücklich direkt im Zusammenhang mit den eingangs erwähnten Computern. Sehr gerne: Crosscompiler, Emulatoren, Game Maker und dergleichen. Auf keinen Fall aber Remakes von Spielen alter Plattformen auf moderner Hardware.

Am 23. April wurde der Spectrum 30 Jahre alt!!! Schade, dass Wolfgang Haller, unser verehrter ehemaliger Klubleiter dieses Jubiläum nicht mehr erleben konnte. Zu dem Ereignis erschienen einige neue Spiele, und auch ich habe zwei „alte“ Spiele geschrieben. Das erste ist eine Umsetzung vom C64, welches sehr gute Bewertungen bekam (aus dem Jahr 1986), das andere ist U-Boot Hunt, das ich in den 90ern angefangen habe. Auch die nächsten Projekte stehen kurz vor Fertigstellung. Diese Spiele waren für die SPC PD-Bibliothek gedacht, aber ich denke dass die meisten User ohnehin von WOS herunterladen.
Ich habe auch bemerkt, dass „Warlord“ welches von Wolfgang geschrieben wurde, eine falsche Datei mit UDG-Zeichen hat. Da ich auf meinen Kassetten mit Sicherheit noch die richtige Datei habe, werde ich eine korrigierte Version erstellen. Keine Ahnung wie das passieren konnte.

Lothar Ebelshäuser fragte an, ob ich nicht eine Mitgliederliste abdrucken könnte. Hier möchte ich sagen, dass ich nicht so sicher bin ob das von den Betroffenen erwünscht ist, wegen des Datenschutzes. Im Prinzip hat Wolfgang ja die Mitgliederliste abgedruckt, allerdings ändern sich die Zeiten und als Pirat bin ich um Datenschutz und Privatsphäre sehr bemüht. Deswegen bitte ich die Mitglieder mir eine E-Mail oder Brief zu schicken ob sie damit einverstanden wären, wenn ihr Name und Kontaktdaten abgedruckt werden, oder nicht.
Ab nächstem Jahr werden wieder die jährlichen Umfragekärtchen gedruckt (falls ich passendes Papier finde), wo diese Frage von Haus aus gestellt wird.
Ein weiterer Wunsch ist ein laminierter Mitgliedsausweis für den Club, mit Foto. Dem wäre ich nicht abgeneigt, allerdings ist das eine Kostenfrage. Besteht ein Interesse? Wie ist Eure Meinung dazu?
Durch die Portoerhöhung der österreichischen Post sind die Versandkosten enorm gestiegen. Das werden sicher schon viele bemerkt haben, als sie die Umschläge ihrer letzten Ausgaben gesehen haben. Ich werde aber trotzdem den Mitgliedsbeitrag nicht erhöhen. Der Club ist wie eine Familie, und da ist Geld nicht das Wichtigste.

LCD-Leszek Chmielewski

Termine 2012

06-08. July 2012

AROK-Party (Ungarn)

http://arok.intro.hu/index2.php?m=4&l=1&p=2

Multiplatform-Treffen: ZX Spectrum, C64, Plus 4, Atari.

Erinnerung:

Hallo Leute!

Der Termin für das Specci-Treffen 2012 in Wittenberg ist Sa./So. 25./26. August. Wir sind wieder im Kulturbund in der Lutherstraße 43a wie 2011. Jemand hatte den Vorschlag gemacht, die Tische ringsum an die Wand zu schieben, damit alles übersichtlicher und mehr Platz wird.
Bis zum nächsten Wiedersehen.

Tschüß,

Norbert Opitz

Mister Beep schickt eine Neuigkeit:
Hello.

Here is a new music-demo for ZX Spectrum 48K:
http://mister_beep.republika.pl/nebula_fight.zip

Music: Mister Beep, code: Shiru, graphic: Trixs.

Please use real hardware (emulators give noisy beeper sound).

Greetings,
MB

Auch Yerzmyey schickt eine Neuigkeit:
Hi there.

Just a short AY song I made for ZX81 demo by Andy Rea.

(The demo works on ZX81 16K with ZonX AY-interface and all others in ZonX standard: Mr X and ZXpand-AY).

The song:
http://chipmusic.org/yerzmyey/music/square-disco-zx81---timex-1000---timex-1500

The demo:
http://www.rwapservices.co.uk/ZX80_ZX81/forums/download/file.php?id=813

Btw, I made a ZX81 SoundTracker instruction, if anybody would be interested:
http://chipmusic.org/forums/topic/6668/soundtracker-zx81-instruction-ayym/

Greetz,
Yerz
http://ay-riders.speccy.cz/

Foreword-English

Welcome to the magazine by users for users. We are primarily dependent YOUR article. I alone can not fill (targeted) 24-32 pages, even I would like to do it. Word of honor! To be sent article, the following rules:

The articles have to deal with the Spectrum, ZX81, SAM Coupe, Sprinter 2000, or close relatives of the Sinclair ZX Spectrum, including articles on appropriate hardware and software are welcome.
MAC / PC software: Only if expressly directly in connection with the above-mentioned computers. I would be very happy about: cross compiler, emulators, Game Maker, and the like. In no case, however, remakes of old games on modern hardware platforms.

On 23 April the spectrum was 30 years

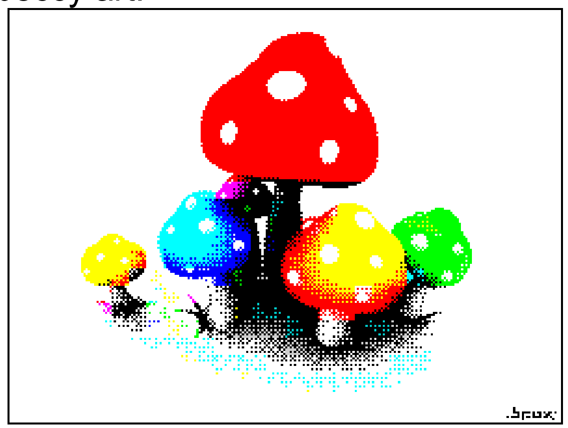
old! Too bad that Wolfgang Haller, our deceased former leader of our club anniversary could not enjoy it.
For the event some new games appeared, and I've written two "old" games. The first is a conversion of a C64 game (from 1986), which got very good reviews, the other is the U-boot Hunt I started in the '90s. The next projects are nearing completion too. These games were destinated for zhe SPC PD library, but I think that most users already download from WOS. I've also noticed that "Warlord" which was written by Wolfgang, has an incorrect file with UDG characters. I still have certainly my tape with the correct file, I will create a corrected version. No idea how this could happen.

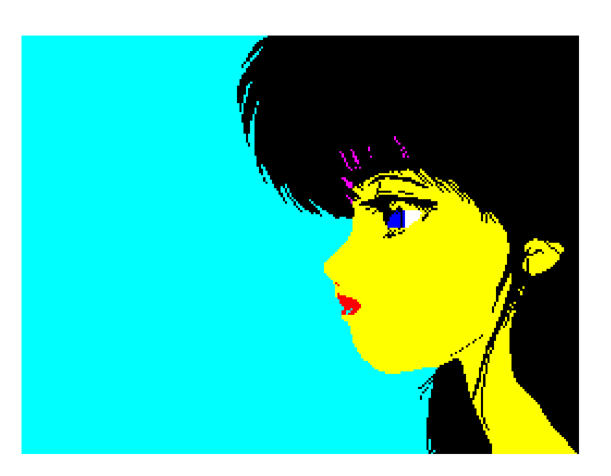
Lothar Ebelshäuser inquired if I could print a list of members. Here I would like to say that I'm not so sure if that is desired by those affected, because of data protection. In principle, yes, Wolfgang has published the list of members, but times change and as a pirate, I'm believing in data protection and privacy very much. So I ask the members to email me or send a letter if they would agree if their name and contact information will be printed, or not.
Starting next year, again printed annual survey cards (if I find a suitable paper), where this question is asked too. Another wish is a laminated membership card for the club, with photo. I would not be averse, however, it is a question of costs. I want to read your opinion.

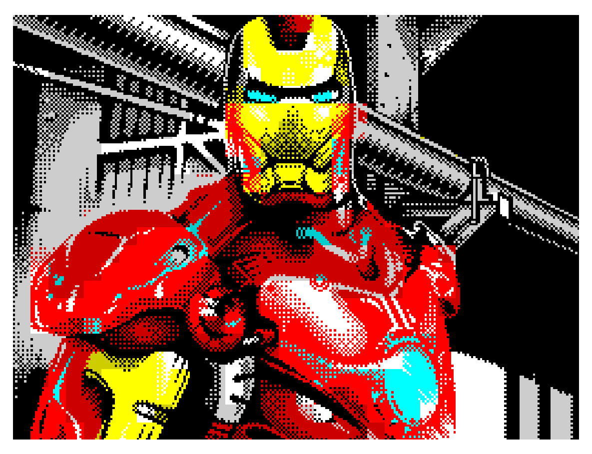
Due to the increase in postage, the Austrian postal shipping costs have risen dramatically. This will certainly have noticed many when checking the stamp on envelopes of the recent issues. But I will still not increase the membership fee. The club is like a family, and money is not the most important here.

LCD Leszek Chmielewski

Speccy art:





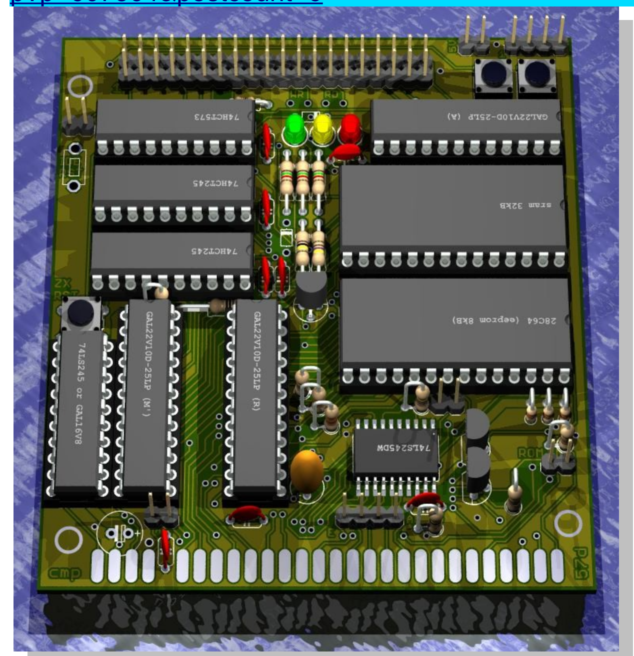


Neuigkeiten für unseren „Alten“

Und wieder habe ich die Ehre euch den neuesten Stuff vorzuführen

Neues DivIDE 57d?

http://www.worldofspectrum.org/forums/showpost.php?p=607364&postcount=6

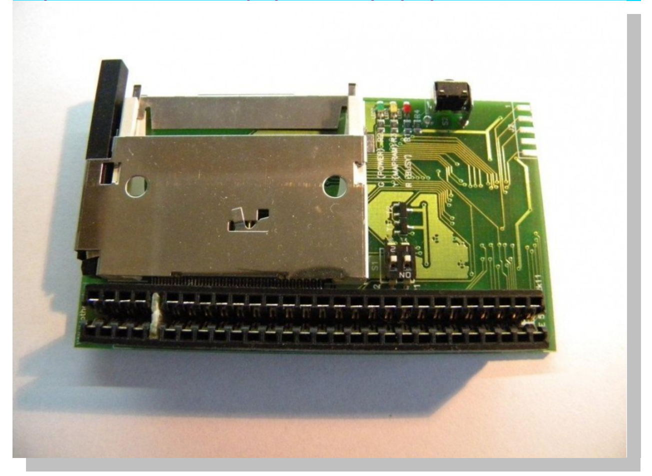


Velesoft hat ein neues DivIDE-Layout vorgestellt, mit auffallenden drei Buttons und zusätzlichen 74LS245 Chips. Trotz Verbesserungen nimmt die Platine genauso viel Platz ein wie die Version c.
Die vorerst für Velesofts private Verwendung gedachte Version weist folgende Verbesserungen auf:

- Schalter für +5 V auf IDE Pin 20 für CFtoIDE Adapter ohne Änderungen.
- Netzanschluss an Bord (3,5 "FDD-Stil) für kleine xxtoIDE Adapter.
- ZX Reset-Taste.
- IDE-Reset-Taster (Reset nur IDE-Gerät - bei SDtoIDE Adapter ist es möglich SD-Karten zu hotswappen, nach dem Tausch der Karte nur IDE-Reset Taste drücken).
- Möglichkeit der Nutzung klassischer oder schlanker SRAM-Speicher-Bausteine.
- Schalter für den Einsatz auf Computern TK90X/TK95 (brasilianischer 60Hz ZX-Klon).
- Verstärkt Signale MREQ, IORQ, RD, WR, RESET, A14, A15.
- Jumper zum Deaktivieren von MREQ
- Hinzugefügt: Bohrungen für Schrauben an Kunststoffgehäuse.
- Möglichkeit mit Schalter umzuschalten auf reinen IDE-Modus (+3e).

Lothareks DivIDE 2k11

http://www.lotharek.pl/product.php?pid=25



Um € 57,80 bekommt man von Lotharek ein vertikales DivIDE mit CF-Slot. Gut, dass es kompakt ist, und einen eingebauten CF-Slot für bis zu 2 Karten hat, schlecht hingegen ist, dass es trotz vertikaler Bauweise keinen durchgeführten Port hat, und alle Chips aufgelötete SMD-Versionen sind (macht sicher Spaß die Dinger zu reparieren). Ich kann nicht unbedingt eine Empfehlung aussprechen da das Gerät auch noch teurer als das normale DivIDE ist. Schade, dass SD-Adapter nicht benutzbar sind.

Feuertempel

http://www.relevovideogames.com/los-templos-de-fuego/?lang=en



Von „Los Templos de Fuego“ („Temple of Fire“) wurde ein erster Screenshot von der MSX-Plattform veröffentlicht. Die Spectrum-Version kann man da ungefähr interpolieren.

Antiquity Jones, nehme ich an!

http://www.worldofspectrum.org/infoseekid.cgi?id=0027506



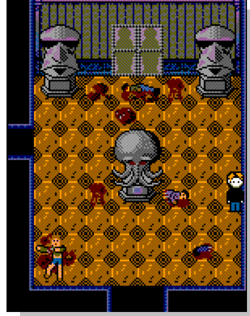
Paul Jenkinson, der bereits viele AGD und ZXBC-Titel geschrieben hat („Broken Friend“, „Kyd Cadet I+II“, „Space Disposal“, „Chopper Drop“), wird nicht müde und veröffentlichte sein neuestes, mit ADG erstelltes Spiel. Dieses erinnert an „Livingstone I presume“. Es handelt sich dabei um einen Arcade-Plattformer, der mit sehr schöner Grafik und Animation aufwartet.

Compiler C-EVO

http://speccy-live.untergrund.net/?p=5191
http://zx.pk.ru/showthread.php?t=18864

Pentagon Evolution hat nun ein SDK bekommen. Es handelt sich dabei um einen C Compiler.





Schon wurden damit ein paar Spiele geschrieben wie XNX (Xonix-Klon) oder Innsmouth (RPG). Vor allem das zweite erregt meine Aufmerksamkeit und könnte der Grund sein sich einen PentEvo zuzulegen.

SAM-XOR

http://cookingcircle.co.uk/



Cooking Circle erschuf den Spectrum Klassiker XOR neu für den SAM Coupé, und es ist eine solide Version geworden, die viel Spaß macht, auch wenn die Grafik hätte ein wenig farbenfroher ausfallen können.

SAM-Disk

http://simonowen.com/samdisk/

Simon Owen hat die Version 3.2 seines Programms SAMDisk freigegeben. Mit diesem Command Line Utility kann man Spectrum- und SAM Disketten lesen oder Images auf Disks schreiben.

ESX-DOS 0.8 erschienen

http://www.worldofspectrum.org/forums/showthread.php?t=38991

Mit der Version 0.8 nähert sich das ESX-DOS seiner Vollendung. Endlich gibt es einen TAP-Browser der mit NMI aufgerufen wird (sieht nicht so gut aus wie FATWARE und zeigt die Titel unvollständig an, aber daran wird noch gearbeitet). Weitere Verbesserungen werden folgen, so soll die Version 0.9 TRDOS emulieren können.
Ach ja, Gasman hat seinen Divideo Player für ESX-DOS angepaßt.

Majikazo ist fertig

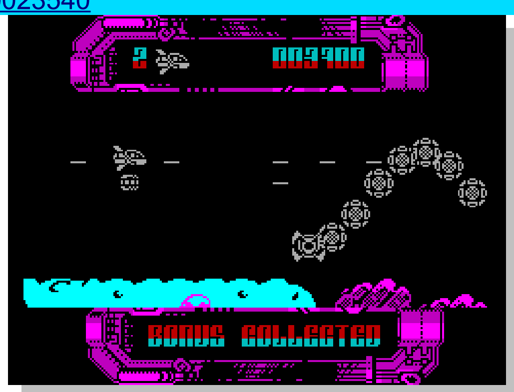
http://www.worldofspectrum.org/infoseekid.cgi?id=0027624



Das MSX Spiel Majikazo ist nun am Spectrum fertig und wurde zum Download freigegeben. Dieses farbenfrohe Spiel erinnert an Solomon's Key und ist wirklich ein Meisterstück. Es gibt zwei Versionen von dem Spiel. Eine für Spectrum 48/128+/+2, und die andere wurde von Metalbrain gefixt, so dass sie auch auf dem Spectrum +2A/+3 lauffähig ist.

Ooops! Tschuldigung!

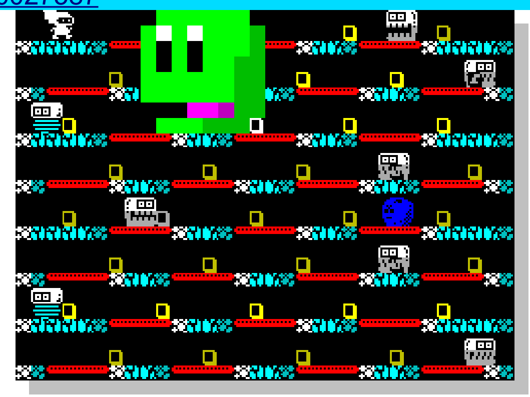
http://www.worldofspectrum.org/infoseekid.cgi?id=0023540



Wenn man in England in Gesellschaft Darmgase hörbar entweichen lässt, sagen die Leute „More Tea, Vicar?“. Jonathan Cauldwell hat sich diesen seltsamen Titel für sein Shoot'em up ausgesucht, welches fast schon als verschollen galt, weil es jahrelang nicht fertig wurde (ein Demo erschien 2004).
Die 128K Version mit einem fetzigen Soundtrack von Yerzmyey ist kostenlos erhältlich, während die 48K Version mit einem anderen letzten Level daherkommt, und von Cronosoft vertrieben wird.
Der Nachfolger soll vertikal scrollen und „Sugar no Milk“ heißen.

Willkommen im Paralleluniversum

http://www.worldofspectrum.org/infoseekid.cgi?id=0027587



„The lost tapes of Albion“ ist ein neues Spiel von R-Tape (Dave Hughes), und es spielt in einem Paralleluniversum wo

es unseren Helden hinverschlagen hat. Beim Transfer verlor er jedoch seine Spectrum-Spiele und muss sie nun wiederfinden (für mich absolut nachvollziehbar). Für jedes Level bekommt er noch ein Spectrumspiel, das es nur im Paralleluniversum gibt. Stichwort: „Horny Horace“.

Und nun das Ganze für den SAM!

http://www.blackjet.co.uk/index.php#lostdisks

Auch den SAM gibts in Paralleluniversen!



Blackjet hat aus „The lost Tapes of Albion“ ein Spiel „The lost disks of SAM“ in Assembler programmiert, und es läuft in Mode 4. Am SAM ist das Spiel schwerer weil das Hauptsprite recht dunkel ist und daher nicht sofort zu sehen, aber trotzdem ist es ein sehr gutes Spiel.

Arcade Game Designer v3.0

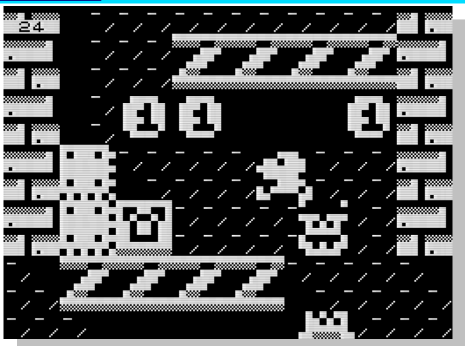
http://www.worldofspectrum.org/infoseekid.cgi?id=0020176



Mit Arcade Game Designer wurden Spiele wie „Kyd Kadet“, „Chopper Drop“ und einige andere geschrieben. Jonathan Coulwell hat nun die dritte Version seines 128K Tools veröffentlicht. Menügesteuertes Inventory ist nur eine der vielen Verbesserungen.

UWOL 81

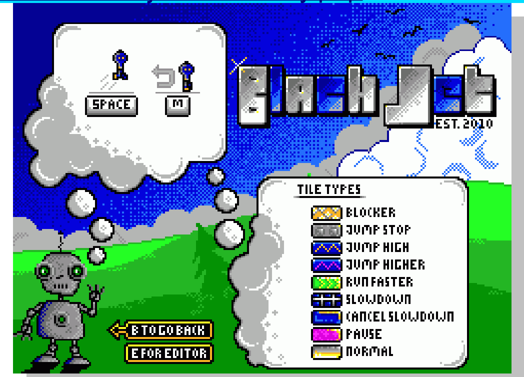
http://www.worldofspectrum.org/forums/showthread.php?t=38486



UWOL – Quest for Money wurde nun für den ZX 81 umgesetzt, und natürlich haben Mojon Twins wieder Sources des Spieles freigegeben.
Die Charmap-Grafik wurde wirklich hervorragend eingesetzt und erlaubt eine sehr gute Spielbarkeit. Meine Meinung: Zehn Sternchen.

Dave Infuriators

http://www.blackjet.co.uk/diary.php



Blackjet arbeitet an dem neuem SAM-Spiel „Dave Infuriators“, einem Plattformer der „anders“ ist. Und zwar wird es nur mit einer „Sprung“-Taste gesteuert, alles andere läuft automatisch ab. Die Figur prallt von Wänden ab um die Richtung zu ändern, alles andere wird von den Plattformen gesteuert.

Snow Bros

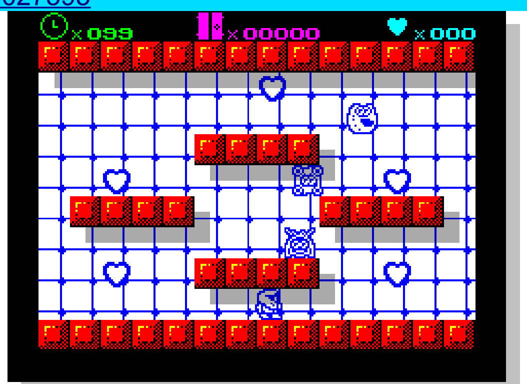
http://www.bytemaniacos.com/?page_id=1968



Climacus, Radastan und Shiru haben das Spiel „Speccy Bros“ endlich fertig gestellt und freigegeben. Es ist eine Interpretation des NES-Hits „Snow Bros“. Die farbenfrohe Grafik verhindert zwar, dass die Sprites pixelweise bewegt werden, aber trotzdem macht das Spiel sehr viel Spaß denn die Entwickler haben seit der ersten Tech-Demo viel Zeit dafür investiert dass die Spielbarkeit stimmt.

Herzensdieb kehrt zurück

http://www.worldofspectrum.org/infoseekid.cgi?id=0027595



Timmy hat sein „Heart Stealer“ ein wenig umprogrammiert und eine „One key“ (nicht zu verwechseln mit dem Begriff 1K, welches für 1 Kb Version stehen würde) gemacht.

Die Figur läuft, und der Spieler betätigt nur die Taste fürs springen. Ist ganz okay, aber die alte Version ist spielbarer.

Das Erwachen

http://www.worldofspectrum.org/forums/showthread.php?t=38673



Konstantyn Dvornik hat sich viel Mühe gemacht um Awaken aus dem Russischen zu übersetzen. Leider existiert das Spiel nur als TRD Version.

Brian vs. The Bullies

http://www.worldofspectrum.org/infoseekid.cgi?id=0027593



Auch mit dem Platform Game Designer kann man ganz nette Spiele schreiben. „Brian vs. The Bullies“ wurde schon 2005 geschrieben und war eigentlich für Cronosoft Cassette 50 geplant, aber anscheinend kamen keine 50 Spiele zusammen, also hat Andrew Potts das Spiel nun freigegeben um 30 Jahre Spectrum zu feiern.

Für die deutschen Emulator User könnte es problematisch werden weil Tasten z und x benutzt werden um die Figur nach links und rechts zu bewegen. Das „z“ ist ja bei deutschen Tastaturen falsch.

Phantomas ins Museum?

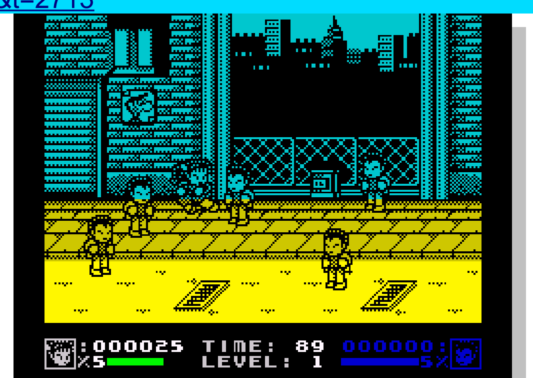
http://www.mojontwins.com/juegos_mojonos/phantomas-en-el-museo-spectrum-48k/



Nein, Phantomas gehört noch nicht zum alten Eisen. Mojon Twins sorgen dafür, dass immer wieder neue Phantomas-Spiele entstehen.
Mit „Phantomas en el Museo“ ist ein Spiel im Auftrag des Museums IAACC Pablo Serrano aus Zaragoza für den Internationalen Museumstag 2012 entstanden. Das Spiel kann man live im Museum spielen, oder wer sich den Weg nach Spanien nicht jedes mal leisten kann wenn er das Spiel spielen will, kann es jetzt herunterladen, und natürlich sind auch Sources verfügbar.

Karate-Zwillinge

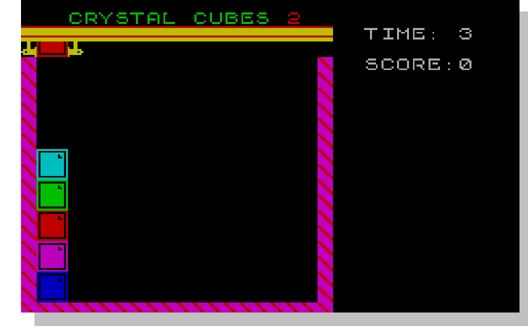
http://www.speccy.org/foro/viewtopic.php?f=6&t=2713



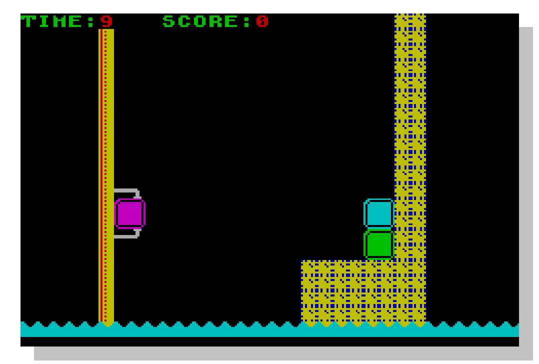
Gandulf arbeitet gerade am Projekt „Karate Twins Story“ welches noch dieses Jahr fertiggestellt werden soll, sobald die Probleme (die in der K.I. des Gegners liegt) beseitigt sind.
Die Manga-Grafik und sanftes Pixel-Scrolling versprechen jedenfalls ein weiteres Highlight für dieses Jahr.

Crystal Cubes

http://www.worldofspectrum.org/infoseekid.cgi?id=0027555
http://www.worldofspectrum.org/infoseekid.cgi?id=0027557



AER haben nach ihrem Puzzler „Crystal Cubes“ nun „Crystal Cubes II“ und „Crystal Cubes III“ freigegeben. Leider wie üblich, als TRD.



Gimme Brighter

http://www.worldofspectrum.org/forums/showthread.php?t=33413

„Gimme Bright“ von Climacus ist jetzt in der Version 2.0 erschienen, und hat neue schönere Grafiken bekommen, die von Radastan gezeichnet wurden, also einem echten Grafik-Profi am Spectrum.
Wer die erste Version hatte, sollte sich die neue Version zumindest ansehen, denn meiner Meinung nach lohnt es sich auf jeden Fall die neue Version herunter zu laden.

Albtraum-Arbeit

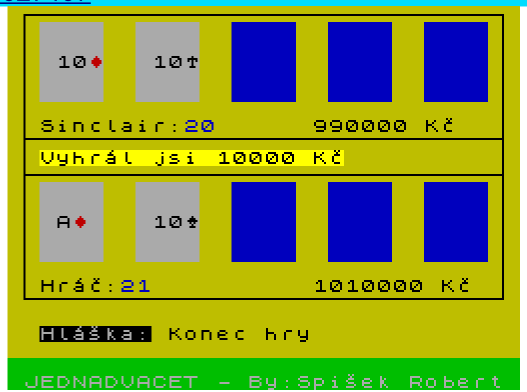
http://www.worldofspectrum.org/forums/showthread.php?t=38428



dm_boozefreak hat einen neuen Manic Miner Klon angekündigt. Sieht zwar nicht schlecht aus. Eines der Screens erinnert an Canabalt.

Eins, zwei, drei...

http://www.worldofspectrum.org/infoseekid.cgi?id=0027407

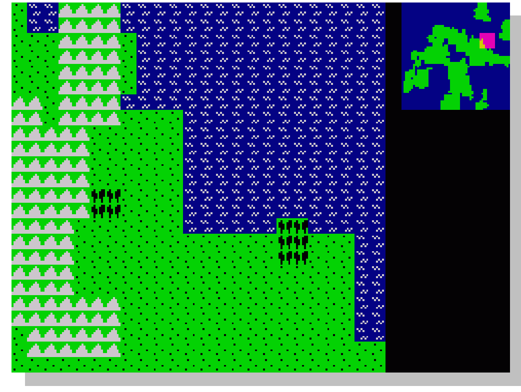


„Jednadvacet" ist ein tschechisches Kartenspiel „Black Jack" welches der Autor 1997 auf seiner Amiga 500 geschrieben hat. Bislang hat er noch nie am Spectrum ein Spiel fertig geschrieben. Nun ist aber der Fluch von Robin Spisek genommen... 2012 ist ein gutes Jahr für neue Spectrum Spiele.

Civilisiert!

http://zx.pk.ru/showthread.php?t=9146

Russische Coder möchten sich erneut an „Civilisation" versuchen, was ja bereits ein mal abgebrochen wurde. Das neue Projekt steht erst am Anfang, aber gut aussehen tut es sicher nicht.



Das Scrolling der Karte ist elendig lahm (da bringe ich sogar in ZXBC ein schnelleres zusammen). Also wenn das nicht die erste Vorschauversion eines WIP wäre, würde ich sagen: Vergessen wir das Ganze. Sorry für die harsche Kritik, aber wenn es erst ab 28 MHz Spectrum spielbar wird, dann nützt es niemanden was. Warten wir es mal ab.

Epic Fail

http://www.worldofsam.org/node/713



Rob Evans (Wubsoft) hat ein neues Spiel für den SAM geschrieben: „Epic Fail". So epic ist der fail aber nicht, denn das flüssige Scrolling im Mode 4 rettet einiges. Das Power up System weiß zu gefallen.

BASIC-Wettbewerb 2012

http://www.bytemaniacos.com/web/?page_id=120

Bytemaniacos haben einen Wettbewerb in BASIC-Programmierung für das Jahr 2012 ausgerufen. Und zwar muss man ein RPG wie Zelda/Final Fantasy oder

Dungeon Master Klon vollständig in BASIC schreiben. ROM-Aufrufe sind erlaubt, aber sonst kein Assembler. Einsendeschluss ist der 30. Dezember, aber die Preise sind sehr klein.

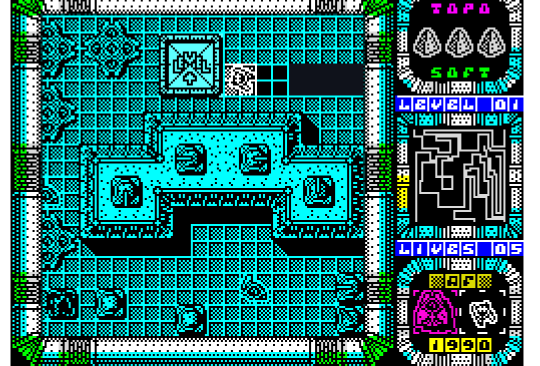
Erudit

http://www.worldofspectrum.org/infoseekid.cgi?id=0027450
http://zx.pk.ru/showthread.php?t=18425

Liebhaber von Scrabble mit russisch-Kenntnissen die einen Spectrum mit TRD Betriebssystem haben, sollten sich Erudit anschauen. Bei dieser Variante kann man anscheinend auch das Wörterbuch editieren.

R.A.M. II

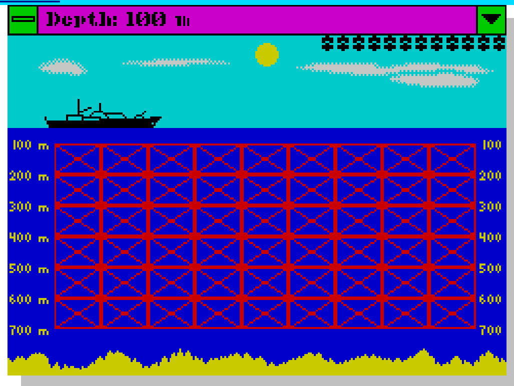
http://computeremuzone.com/ficha.php?id=915



Wenn das Spiel gefunden wird, könnte es interessant werden! Erst vor kurzem sind die Screenshots zu R.A.M. II aufgetaucht, aber wenn Topo 21 das Spiel nicht fertiggestellt hat, woher kommen denn die Screenshots? Möglicherweise existiert ein Preview?

Auf zur Jagd

http://www.boriel.com/forum/gallery/u-boot-hunt-t767.html



LCD (ja, das bin ich) hat sein neuestes Spiel fertiggestellt: „U-Boot Hunt“. Wie erwartet, in ZXBC geschrieben, muss man fast blind ein Nazi U-Boot versenken. Man weiß nur wie die Entfernung zwischen Explosion und U-Boot war, aber alles andere bleibt verborgen. Und als das noch nicht genug wäre, bewegt sich das U-Boot kontinuierlich nach links oder rechts.
Dieses Projekt wurde in den 90ern angefangen und war längere Zeit verschollen. Auf meiner Homepage war mal ein Screenshot zu sehen. Die für ZXBC angepasste Version ist schneller und hat einige besondere (zumindest für mich) Effekte.

Kampf den Killermollusken!

http://www.boriel.com/forum/gallery/new-game-earthraid-t766.html

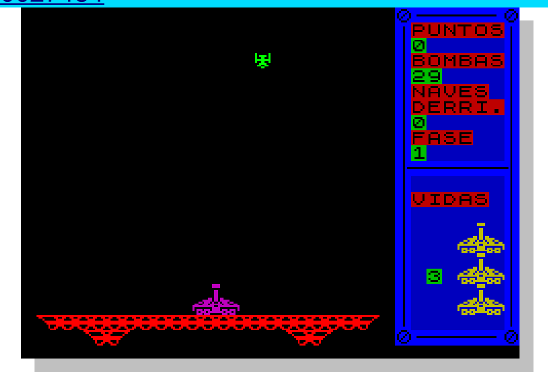


In der Happy Computer 8/86 erschien das Listing „Earthraid“ für den C64.

Dieses hat LCD (ja, das bin erneut ich) nie losgelassen. Er wollte unbedingt eine Spectrum-Version dieses Strategiespieles schreiben. Und nun ist es soweit. Das Spiel ist fertig. Geschrieben mit Boriels ZXBC präsentiert es sich wie auf dem C64, jedoch mit einigen Unterschieden. Ja, richtig gelesen! Gegenüber der C64 Version wurde es ein wenig verbessert und erweitert. Und IMHO ist es eine sehr gute Version geworden.

Galaktron

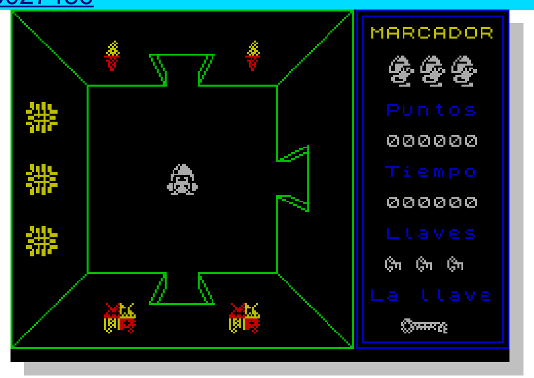
http://www.worldofspectrum.org/infoseekid.cgi?id=0027484



Carlos David Diaz hat 1987 begonnen „Galaktron“ zu entwickeln. Nun ist es fertiggestellt und auf WOS zu haben. Natürlich hat es nicht 25 jahre gedauert bis das Spiel fertig war, sondern es waren ein paar Jahre Unterbrechung dazwischen. Nette Beeper-Tunes!

Vampire Atac

www.worldofspectrum.org/infoseekid.cgi?id=0027486



Atic Atac stand Pate für „Vamperi“, doch das Interessante ist, dass es in BASIC geschrieben wurde. Das Tempo leidet zwar stark, aber trotz der Tatsache, dass das Spiel 2012 geschrieben wurde, fühlt man sich nach 1984 zurückversetzt. Einfach wunderbar!

Titan-Adventure

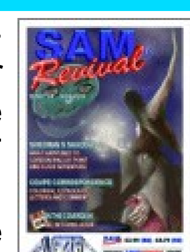
http://www.worldofspectrum.org/infoseekid.cgi?id=0027485



„Regreso a Titan“ ist ein weiteres neues Spiel in BASIC, das von Carlos geschrieben wurde. Die Entwicklung begann 1986. Es fehlen mir leider Spanischkenntnisse um das Spiel zu genießen.

SAM Revival Neuigkeiten

http://www.samcoupe.com/



SAM Revival Ausgabe 24 steht kurz vor ihrer Fertigstellung. Eine Vorschau ist als PDF herunterladbar.
Wie andere Postunternehmen erhöht auch Royal Mail ihre Preise kräftig, wodurch die Subskriptionspreise stark ansteigen werden.
Ausgabe 24 kostet 4,70 Pfund mit Versand (EU).
Das Problem habe ich auch hier mit österreichischer Post im größeren Ausmaß vor kurzem selbst gehabt.

USB-Schnittstelle und USB-Stick am ZX Spectrum

von ZX-Heinz

Kürzlich hatte ich mir ein divIDE zugelegt, nun geht das Laden von Dateien, die vorher im TAP-Format mittels PC auf die Speicherkarte aufgespielt wurden, blitzschnell. ABER: Ich kann weder speichern, noch verträgt sich das divIDE mit dem Interface 1 (IF1) am Spectrum. Zur Überwindung dieser Probleme und angeregt durch die in Mahlerts beim ZX-Treffen gesehenen USB-Sticks am ZX81 habe ich mich selbst auch mit einer Entwicklung beschäftigt.
Zwei Grundprinzipien bestimmten dabei mein Vorgehen:

1. Minimale Hardwareerweiterungen unter Verwendung des IF1.
2. Weitgehende Nutzung der im ZX-ROM und im IF1-ROM vorhandenen Routinen im Geiste von Sir Sinclair.

Zwei Projekte wurden inzwischen verwirklicht bzw. sind noch in Entwicklung.

USB-Interface zum direkten Datenaustausch mit dem PC

Das IF1 bringt eine RS232-Schnittstelle mit, aber Vorsicht, die 9-polige IF1-Buchse hat nicht die Standardbelegung von COM-Schnittstellen!! Also ein Extrakabel anfertigen! Der Datentransfer geschieht dann unter WINDOWS z.B.mit dem Programm Hyperterminal, das Empfangen, Senden und Speichern von Datenfiles zulässt.

Angenehmer als eine COM ist aber eine USB-Verbindung! Die Firma ELV bietet dazu einen **Mini-USB-Modul UM2102** für unter 6€ an. Komplettiert mit einem 9-poligen Stecker für das IF1 und einem Umsetzer MAX232 von RS232-Pegel auf TTL-Signale erhält man damit schon die gewünschte Verbindung zum PC, wieder über Hyperterminal (Siehe Abb. 1, oberer Baustein).
Damit steht die gesamte Fileverwaltung des PC zur Verfügung. Aber mehr noch: Über die IF1-Befehle SAVE*"b", SAVE*"b" CODE bzw. SAVE*"b"SCREEN$ und die entsprechenden Ladebefehle hat man den gesamten PC als Speichermedium zur Verfügung!! Also Hunderte von Gigabytes nebst bester Fileverwaltung!

USB-Stick am Speccy

Autonom sollte der Speccy aber auch sein können, nicht immer will man den PC mitschleppen. Also braucht man eine Speichermöglichkeit auf USB-Sticks. Die ZX81-Freunde benutzen hierzu vielfach den VDRIVE2-Modul von Vinculum mit dem VNC1-Controller (näheres z.B. in http://docs-europe.electrocomponents.com).
Das hieß für mich: Ein VDRIVE2 bestellen, 9-poligen Stecker und MAX232 dazwischen und fertig (s. Abb. 1, unterer Teil). Von Thomas hatte ich in Mahlerts gelernt, dass dieser Controller durch eine Befehlsliste gesteuert wird, und sein Assemblerprogramm zur Realisierung eines kleinen DOS oder Monitors auf dem ZX81 half mir ebenso sehr weiter (vielen Dank für beides!!).
Nun zum Betrieb mit dem Speccy. Das IF1 bietet mir der seriellen Schnittstelle bereits eine sehr einfache Möglichkeit,

mit dem USB-Stick zu kommunizieren. Bisher habe ich die wichtigsten Funktionen wie DIR, WRITE, READ, ERASE und SAVE/LOAD zu Demozwecken und zum Lernen mit einem BASIC-PROGRAMM gesteuert (s. Listing). Das ist der Stand der Dinge, und mit der Hilfe weiterer Clubfreunde wird daraus schon bald eine wirklich praktikable, feine und sehr kompakte Treibersoftware entstehen.

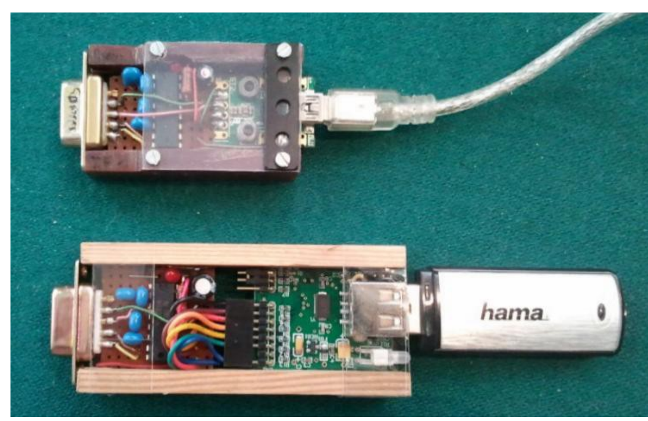


Abb. 1

```
 90 REM LISTING VD08 Kann Schreiben&Lesen und SAVE&LOAD-CODE
 100 REM VDRIVE2-TEST
 101 CLS
 102 PRINT "B(ASIC)"+CHR$ 13+"D(ir)"+CHR$ 13+"R(ead)"+CHR$ 13+ "S(chreiben)"

+CHR$13+"E(rase)"+CHR$ 13+"(SAVE-)C(ODE)"+CHR$ 13+"L(oad CODE)"
 104 INPUT "Auswahl?";I$
 105 CLS
 106 IF I$="R" THEN  GO TO 190
 108 IF I$="D" THEN  GO TO 300
 110 IF I$="S" THEN  GO TO 500
 115 IF I$="E" THEN  GO TO 800
 117 IF I$="C" THEN  GO TO 900
 118 IF I$="L" THEN  GO TO 700
 120 PRINT CHR$ 13;"ENDE"
 125 STOP
 130 PAUSE 0: GO TO 100
 190 REM LESEN
 192 INPUT "FILENAME=";L$: PRINT "Los!";
 193 PRINT #5;"RD ";L$;CHR$ 13;
 194 LET X$=INKEY$#5: PRINT "RD CHR13 verwerfen"
 195 GO SUB 200: GO TO 130
 197 REM
 200 REM LESE-ROUTINE
 210 LET a$=INKEY$#5
 220 LET a=CODE a$
 230 PRINT a$ AND a>=32;" "+STR$ a+" " AND (a>0 AND a<32 AND a<>13);a$ AND a=13;
 240 IF a$=">" THEN PRINT INKEY$ #5: RETURN
 250 GO TO 200
 300 REM DIR
 310 PRINT #5;"DIR";CHR$ 13;
 320 GO SUB 200
 330 GO TO 130
 500 REM SCHREIBEN
 510 INPUT "FILENAME=";F$
 520 PRINT #5;"OPW "+F$;CHR$ 13;
 525 PRINT F$;" open"
 530 GO SUB 200
 550 INPUT "Zu schreibender Text=";T$
 560 PRINT #5;"WRF ";CHR$ 0;CHR$ 0;CHR$ 0;CHR$ LEN T$;CHR$ 13;
 570 PRINT 570: REM  GO SUB 200
 580 PRINT #5;T$;
 585 PRINT "Daten gesendet"
 586 GO SUB 200
 600 PRINT #5;"CLF "+F$;CHR$ 13;
 605 PRINT "File closed"
 610 GO SUB 200
 620 GO TO 130
 700 REM  LOAD-CODE
```

```
 710 PRINT #5;"RD ";"1";CHR$ 13;
 712 LET X$=INKEY$#5: PRINT "RD
CHR13 verwerfen"
 715 PRINT "DATEN KOMMEN,
MUESSEN VERARBEITET WERDEN"
 720 LOAD *"b"CODE 40000
 730 GO SUB 200
 740 GO TO 130
 800 REM ERASE/DELETE
 810 INPUT "FILENAME=";E$
 820 INPUT "Wirklich loeschen
1/0?";E
 825 IF E=1 THEN  PRINT #5;"DLF
"+E$;CHR$ 13;: GO SUB 200
 830 GO TO 130
 900 REM SAVE-CODE 40000,10
 920 PRINT #5;"OPW "+"1";CHR$
13;
 930 PRINT "File open"
 935 GO SUB 200
 940 PRINT #5;"WRF ";CHR$ 0;CHR$
0;CHR$ 0;CHR$ (9+10);CHR$ 13;
 960 SAVE *"b"CODE 40000,10
 962 PRINT "Daten gesendet"
 963 GO SUB 200
 965 PRINT #5;"CLF "+"1";CHR$
13;
 967 PRINT "File closed"
 970 GO SUB 200
 980 GO TO 130
1000 REM INITIALISIERUNG
1005 FORMAT "b";9600
1010 OPEN #5;"b"
1015 PRINT #5; CHR$ 13;: GOSUB
200: REM Warten auf Firmware-
Meldung
1020 STOP
```

ZX-Heinz

MC-Verschiebung

Hallo Spectrum-Freunde!!

Heute wollen wir uns mal mit dem leidigen Thema der frei verschiebbaren Maschinenprogramme beschäftigen. Leidig deshalb, weil es im Vorfeld eine Menge zu beachten gibt, wenn man solche Maschinenroutinen verfasst. Wie bekannt sein dürfte, gibt es ja Anpassprogramme aber man sollte vielleicht gleich bei der Programmierung darauf achten, dass die Programme an keine bestimmte Stelle im Speicher gebunden sind. In allgemeinen ist ein längeres Z 80-Programm wegen seiner im Code vorkommenden absoluten Adressen an eine feste Laufadresse gebunden. Diese absoluten Adressen tauchen vor allem in Lade- und Sprungbefehlen, aber auch in Unterprogrammaufrufen auf. Ziel ist es also, diese absoluten Adressen aus dem Programm zu verbannen. Recht einfach gestaltet sich dies bei den Ladebefehlen, da diejenigen Ladebefehle, die auf Speicherbereiche zielen, die nicht zum Programm gehören, nicht verändert werden müssen. Wir müssen nur die Anweisungen beachten, die direkt auf Daten innerhalb des verschiebbaren Programmcodes zugreifen. Hierfür machen wir uns zunutze, dass die USR-Funktion dem aufgerufenen Maschinenprogramm die eigene Startadresse im BC-Register zur Verfügung stellt.

Gleich am Beginn unseres Programmes retten wir deshalb den Inhalt des BC-Registers an einen sicheren Ort z. B. in eine BASIC Systemvariable, von der wir wissen, dass sie im Programmablauf nicht verändert wird. Um jetzt auf Daten

innerhalb des Programmes zugreifen zu können, muss die Adressbestimmung wie folgt vor sich gehen: Die Adresse errechnet sich als Summe aus der geretteten Startadresse und der programmrelevanten Adresse der Daten (dies ist gleich der Abstand zur Startadresse). Sofern der Datenbereich klein ist, mag es auch genügen, am Programmanfang die Bereichsadresse in das X-Register zu bringen (wieder über den Mechanismus mit der Startadresse in BC und der programmrelativen Adresse). Somit sind dann alle kritischen Ladeanweisungen X-indiziert durchgeführt. Zeigt X genau auf die Mitte des Datenbereichs, hat man Zugriff auf 256 Bytes (IX-128 bis IX+127), ohne dass man das Indexregister verändern muss. Nahezu hoffnungslos wird das ganze Vorhaben, wenn man einen JP-Befehl, der eine weite Distanz überbrückt, durch relative Sprünge ersetzen will. Dazu bräuchte man eine Unmenge von JP-Anweisungen, die das Programm sehr unübersichtlich und groß werden ließen. Ein CALL kann man in der Weise überhaupt nicht simulieren. Es gibt hier zwei Lösungsvorschläge, die wir uns ansehen wollen, jeder mit Vor- und Nachteilen. Die am Ende abgedruckten Assemblerprogramme zeigen beide Ansätze. Der erste Weg ist sehr kurz aber er hat unschöne Unterprogrammaufrufe. Das Programm schreibt zu Beginn eine kleine Hilfsroutine in den Bereich MEMBOT. Natürlich könnte auch jeder andere Speicherbereich benutzt werden, solange seine Adresse für das Programm konstant ist. Jeder Unterprogrammaufruf und jeder absolute Sprung läuft dann über diese Routine, indem die programmrelative Adresse des Zielpunktes in HL geladen wird und dann ein CALL MEMBOT oder JP MEMBOT ausgeführt wird. Aufgabe der Routine ist es dann, die Zieladresse zu berechnen und den Sprung dorthin auszuführen. Wer sich ein wenig mit Maschinensprache auskennt wird feststellen, dass diese Methode nur sehr wenige Kommandos benötigt und deshalb sehr schnell ist. Allerdings hat sie auch Nachteile. Durch die Benutzung eines Registerpaares steht dieses nicht für die Parameterübergabe zur Verfügung. Außerdem benötigt jeder Sprung 6 Bytes was dieses Verfahren somit nur für solche Programme interessant macht, die nur wenige zur ersetzende JP und CALL Anweisungen haben.

Die andere Lösung benötigt für einen CALL nur 2 Bytes ist dafür aber langsamer. Jeder CALL einer programmeigenen Routine wird durch das Kommando RST $10, gefolgt von einem Byte, ersetzt. Dieses Folgebyte gibt die Nummer des gewünschten Unterprogrammes an. In der Reihenfolge der Numerierung sind die programmrelativen Anfangsadressen der Unterprogramme in der Liste UPTABL eingetragen. Bekannterweise benutzt das Spectrum-OS normalerweise RST $10 für die Ausgabe von Zeichen über irgendeinen Strom. Der Inhalt der Systemvariablen CURCHL gibt die Adresse der Kanalbeschreibung an, die mit dem gerade geöffneten Strom verbunden ist. Die Adresse der Output-Routine, die zum Kanal gehört, ist in den ersten beiden Bytes enthalten. Und genau dorthin verzweigt der RST $10 Befehl. In unserem Programm wird nun der Pointer CURCHL so umgebogen,

dass er zu der eigenen UPCALL Routine zeigt. UPCALL korrigiert zuerst die Register und Stacksituation, die durch die gerade durchlaufene ROM-Routine etwas durcheinander geraten ist. Die Output-Routine arbeitet nämlich mit dem zweiten Registersatz und HL' wurde auf den Stack zwischengespeichert. UPCALL liest dann die Nummer des gewünschten Unterprogrammes und holt sich die zugehörige relative Adresse aus der Liste UPTABL. Dazu addiert sie die Adresse START und verzweigt schließlich zu der echten Adresse des Unterprogrammes. Damit wird immer ein CALL zu der gewünschten Adresse ausgeführt aber kein JP. Natürlich könnte man UPCALL so verändern, dass z. B. zu den Adressen 0 bis 19 immer ein JP und zu den anderen immer ein CALL ausgeführt wird. Oder aber man löscht am Zielpunkt einfach die überflüssige Rücksprungadresse auf dem Stack.

Die nachfolgenden Assemblerlistings zeigen einige Beispiel-UPs, um noch einmal das Gesagte zu verdeutlichen. Die Programme wurden von Michael Schramm entwickelt.

Eine Anmerkung zum Schluß: Die folgenden Programmlistings wurden von Originalassemblerlistings eingescannt. Trotz sorgfältiger Durchsicht können Transferfehler nicht ausgeschlossen werden. Sollten sich welche eingeschlichen haben, bitten wir um Information im Clubheft oder direkt an uns.

Soviel für heute. Bis bald hier im Info!!

©2004/2011 by Harald R. Lack, Möslstraße 15 a, 83024 Rosenheim und Hubert Kracher, Schulweg 6, 83064 Raubling

----

Listing 1

```
;Frei verschiebbares Programm
;einfache, schnelle Version
;Unterprogrammaufruf speicheraufwendig
;
MEMBOT = 23698
OPEN = $1601
;
            ORG 30000
            ;Startadresse beliebig
START       LD HL,MEMBOT  ;Im Bereich MEMBOT
            LD (HL),$C5   ;den unten aufgeführten
            INC HL
            ;Maschinencode erzeugen
            LD (HL),$01
            INC HL        ;BC enthält die
            LD (HL),C     ;Adresse START (das bringt
                          ;der USR
            INC HL
            ;Mechanismus mit sich)
            LD (HL),B
            INC HL
            LD (HL),$09
            INC HL
            LD (HL),$C1
            INC HL
            LD (HL),$E9
;
TEST        LD HL,UP1-START;Aufruf eines
                           ;Unterprogrammes
            CALL MEMBOT    ;mit relativer
                           ;Adressangabe
            LD HL,ENDE-START ;und ein entsprechender
                           ;Sprung

            JP MEMBOT
Up1         LD A,2         ;Strom 2 öffnen
;
            CALL OPEN      ;(Ausgabe auf dem
                           ;Bildschirm)
            LD A," A"
            RST $10        ;und ein paar Buchstaben
                           ;drucken
            LD A,"B"       ;als Beweis dafür, dass
                           ;dieses
            RST $10
            ;Unterprogramm erreicht wird
            LD A, $0D
            RST $10
            RET
ENDE        LD A,"X"       ;auch hier einige
                           ;Buchstaben ausgeben
            RST $10
            LD A,"Y"
            RST $10
            LD A,$OD
            RST $10
            RET
            END
```

Listing 2

```
;Die folgende Routine wird im Bereich MEMBOT
```

```
generiert
;
;PUSH BC
;BC-Inhalt retten
;LD BC,START
;Mit Hilfe der eingepokten
;ADD HL,BC
;Adresse START aus der relativen die
;POP BC
;absolute Adresse berechnen
;JP (HL)
;Sprung an die gewünschte Adresse

;aufwendigere Version mit sehr kurzem
;Aufrufmechanismus (2 Byte je Aufruf)

ERRSP = 23613
CURCHL = 23633
BREG = 23655
HELPCH = 23728
;
ORG 40000                   ;beliebig
;
START       LD HL,OFFSET+1-START
            ADD HL,BC       ;In das Unterprogramm
            LD (HL),C       ;UPCALL die akt.
            INC HL          ;Adreßdifferenz
            LD (HL),B       ;einschreiben
            LD HL,UPCALL-START
            ADD HL,BC       ;Adr. UPCALL in
            LD (HELPCH),HL  ;HELPCH.
            LD HL,HELPCH
            LD (CURCHL),HL
            LD HL,UPTABL-START
            ADD HL,BC
            EX DE, HL
            LD HL,TABADR+1-START

            ADD HL,BC       ;Adresse UPTABL
            LD (HL),E       ;errechnen und
            INC HL          ;in das Unterprogramm
UPCALL
        LD (HL),D           ;einschreiben.
        RST $10             ;BEEP-UP
        DEFB 1              ;aufrufen.
;
;
        RST $10             ;PRINT-UP.
        DEFB 0
        DEFT "Test-Text"
        DEFB 13             ;CR
        DEFB 255            ;Endmarke.
        RST $10             ;Noch ein BEEP.
        DEFB 1
        LD SP,(ERRSP)       ;Stack löschen und
        JP $1B76            ;Sprung in den BASIC-
                            ;Interpreter

UPTABL      DEFW PRNTST-START   ;UP 0
        DEFW BEEP-START ;UP 1
        DEFW PRINT2-START ;UP 2
        DEFW PRINT3-START ;UP 3

UPCALL
        LD (BREG),A         ;Akku merken.
        POP HL              ;Rücksprungadresse
                            ;löschen.
        POP HL              ;Alter Wert für HL´
        EXX                 ;wieder 1. Regset
        EX (SP),HL          ;UP-Nummer-Byte
        LD A,(HL)           ;in den Akku.
        INC HL              ;Rückspungadresse
        EX(SP),HL           ;erhöhen.
            PUSH HL         ;HL retten.
            LD H,0          ;Akkuinhalt ver
            SLA A           ;doppeln und in HL
            RL H            ;übertragen, dann
            LD L,A          ;A wiederher-
            LD A, (BREG)    ;stellen.
            PUSH DE         ;Echte Adresse
TABADR      LD DE,0         ;in UPTABL
            ADD HL, DE      ;errechnen,
            LD E,(HL)       ;Tabelleneintrag
            INC HL          ;lesen und
            LD D,(HL)       ;Offset dazu zählen
OFFSET      LD HL,0
            ADD HL,DE       ;Alte Registerinhalte
            POP DE          ;wiederherstellen
            EX (SP),HL      ;und Sprung in das
            RET             ;gewünschte Unterprogramm.
;
PRNTST      EX (SP),HL      ;Dieses Unterprogramm
            LD A,(HL)       ;druckt den
            INC HL          ;Text, der
            EX (SP),HL      ;hinter dem
            CP 255          ;Unterprogrammaufruf
            RET Z           ;steht.
            RST $10         ;255-Byte ist
            DEFB 2          ;Endemarke.
            JR PRNTST
;
BEEP        PUSH AF
            PUSH BC         ;Aufruf der
            PUSH DE         ;BEEP-Routine
            PUSH HL         ;im ROM.
            LD DE,100       ;DE bestimmt die
            LD HL,200       ;Tonlänge,
            CALL $03B5      ;HL die
            POP HL          ;Tonfrequenz.
            POP DE
            POP BC
            POP AF
            RET
;
PRINT2      PUSH AF         ;PRINT CHR$ Akku;
            LD A,2
            JR OFEN
;
PRINT3      PUSH AF         ;LPRINT CHR$ Akku;
            LD A,3
OPEN    EX (SP),HL
        PUSH HL
        LD HL, CURCHL)
        PUSH HL
        PUSH DE
        PUSH BC
        CALL $1601          ;OPEN-Routine
        POP BC              ;Im ROM,sie
        POP DE              ;setzt CURCHL.
        POP HL
        POP AF
        RST $10             ;Zeichen ausgeben
        LD (CURCHL),HL      ;Alten Wert von
        POP HL              ;CURCHL wiederher-
        RET                 ;stellen für den
                            ;Unterprogrammaufruf-
        END                 ;mechanismus
```

English summary

If you are writing programs in machine

code, there is often the problem, that programs are bound to a special address in memory. This brings us to the problem, that programs can not be moved in memory without making code inoperable. In our today´s article, we will give some tips, how you can make your machine code more flexible, means here we show you some hints to take static calls and jumps and replace them by more dynamic functions. Maybe it will help you to make some progress in writing relocatable code. Two example routines, to show you how it could work, are listed at the end of the article.

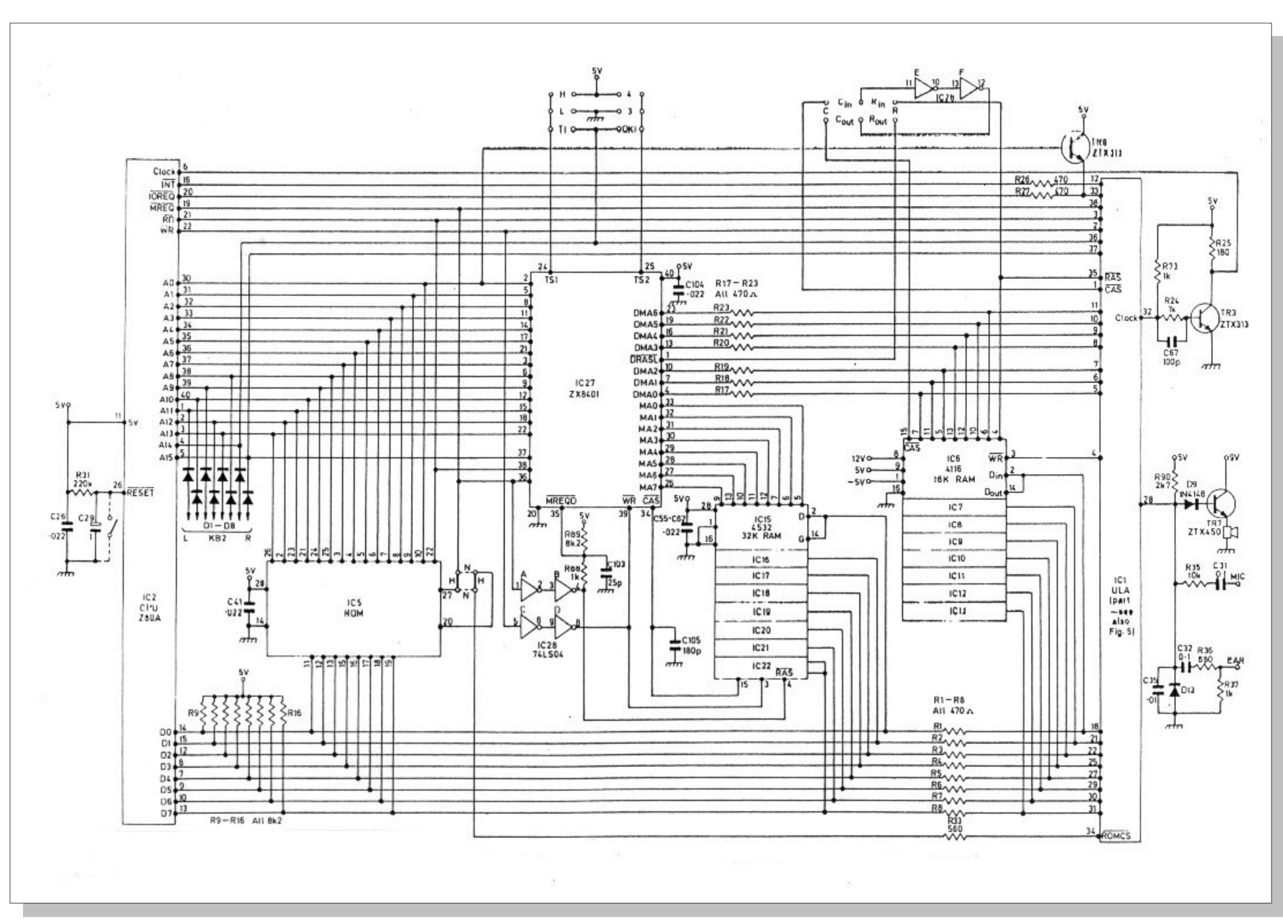
Servicing Sinclair Computers Part 6

http://www.worldofspectrum.org/hardware/rep7.html

continued, and issues 4A, 4B, 5 and 6A subsequently appeared. With each new version the board varies somewhat from the previous one. The differences with the later versions are highlighted below. We'll deal with the voltage generator section separately because although the basic design remained the same there were a number of modifications.

Computing circuitry used in the issue 6 version of the Sinclair Spectrum microcomputer.
Pins 7/11/5/13/12/10/6 of IC6 are connected to chassis via 1k resistors (R50-56)

The Later Spectrums



Previous instalments in this series have dealt with versions of the Spectrum up to and including the 3B. Development

The issue 4A and 4B boards a re similar to the 3B, the main exception being the use of a 6C001-7 ULA chip. This

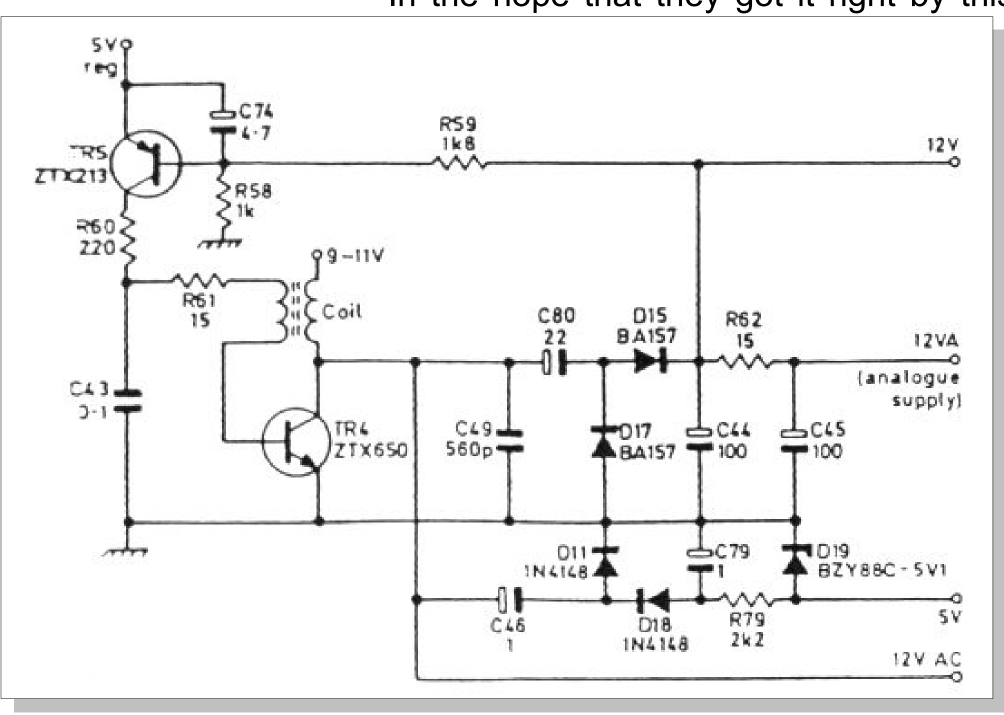
necessitated a timing modification. The two spare nand gates in IC24 are connected in series (to maintain the correct polarity) and replace R32 in the RAS line to IC3 and 1 C4 (see Fig. 1 part 3). Replacement ULAs must be of the -7 or later type. A major change was introduced with the issue 5 board. The six decoder/multiplexer chips IC3, IC4, IC23, IC24, IC25 and IC26 were replaced with a Mullard ULA type ZX8401. But something seems to have gone wrong somewhere because the two gates are still needed in the RAS Line and the new circuit requires an additional four. A 74LS04 hex inverter chip (IC28) provides the six inverters required. Although these changes greatly altered the appearance of the board the basic circuitry hasn't changed very much and servicing shouldn't be affected.

The issue 6 version is very similar to the issue 5 but there's now an alternative supplier of the main ULA (IC1) - Saga joins Ferranti. Certain component changes go with this (see Table 4). Fig. 12 shows the computing circuitry used in the issue 6 version - refer to Fig. 5 for the rest of the circuit. Note that for clarity some supply line decouplers have been omitted, also the connections to the edge connector (refer to Table 3 for these)

The Voltage Generator Circuit

Throughout the development of the Spectrum circuit that's been most subject to change has been the voltage generator. The issue 3/3B version shown in Fig. 4 had already been substantially modified from the issue 1 version. The item that's seen most alteration has been R60, whose value has gone up and down in an almost random manner. Some of the changes are more logical. For example the introduction of the 22uF capacitor (C80) in the 12V supply: this isolates the 9-11V input from the output when there's an oscillator fault. It seems that each time the board was changed this circuit was subject to modification whether or not it improved the performance.

In the hope that they got it right by this



The issue 6 voltage generator circuit.

stage the issue 6 circuit is shown in Fig.

13 (lucky for some?). If you are making any changes I would recommend that you use the values shown here, though if you compare this circuit with Fig. 3 you'll see that they differ in only a few respects. In any case, when TR4 has blown the minimum alteration I'd advise would be to change R60 to 220ohm and to fit C80 and D17 if these are not already present. Any other changes are up to you.

Correction

Note that the EAR socket circuit was shown incorrectly in Fig. 5. It should be as shown in Fig. 12 in all versions.

Alternative Components

Instructions for fitting alternative ROM and ULA chips have been given in earlier instalments. RAM chip replacements have also been dealt with. It's recommended that all replacement ceramic capacitors are of the axial type - if you can get them. Here's a list of alternatives to the E-line transistors (you may find that some are more obscure than the originals):

Table 4: Component changes when alternative ULAs are fitted to issue 8 boards.

Component	Saga ULA	Farranti ULA
C30	Delete	22nF
C32	220nF	100nF
C35	100nF	10nF
C52/3	Delete	150pF
C67	Delete	100pF
C68-71	Delete	100nF
C76	Delete	22nF
R33	560ohm	680ohm
R34	Delete	15ohm
R48	Zero ohms	2k2
R49	Delete	10k
R50	Delete	4k7
R51	1k2	2k2
R56	Delete	1k
R63	Delete	1k
R65-67	Delete	10k
R68	Delete	6k8
R69	Delete	10k
R73	Delete	1k
R74-5	Delete	10k
R76-7	Delete	1k

.

Original Transistors	Alternatives
ZTX650	TIPP31, ZTX651 MPS2369,
ZTX313	MPS2713
ZTX213	BC213, BX214

Source: Television Magazine 1986
Jim Grimwood

256Kb Speichererweiterung für Spectrum 16/48K

Speichererweiterung um einen Spectrum 48K in 128K zu verwandeln, das gibt es zwar, siehe da:
http://hw.speccy.cz/128krebuild.html
Aber es gab mal in Happy Computer (leider unvollständige Schaltzeichnung) eine Speichererweiterung für Spectrum 16K um ihn mit zusätzlichen 256 Kb auszustatten (Insgesamt also 272 Kb). Diese ist leider mit dem Spectrum 128K inkompatibel da eine andere Umschaltadresse verwendet wird, kein zweiter Bildschirmspeicher zur Verfügung steht, und die Bänke 32 statt

16 KB haben. Trotzdem ist diese äußerst interessant gelöst.
Ich wurde gebeten mal diese 256 KB Speichererweiterung zu beschreiben, und da ich aus sammlungstechnischen Gründen eine derartige Speichererweiterung besitze (war sehr teuer weil selten!), gehe ich diesem Wunsch hiermit nach.

256k-Speichererweiterung

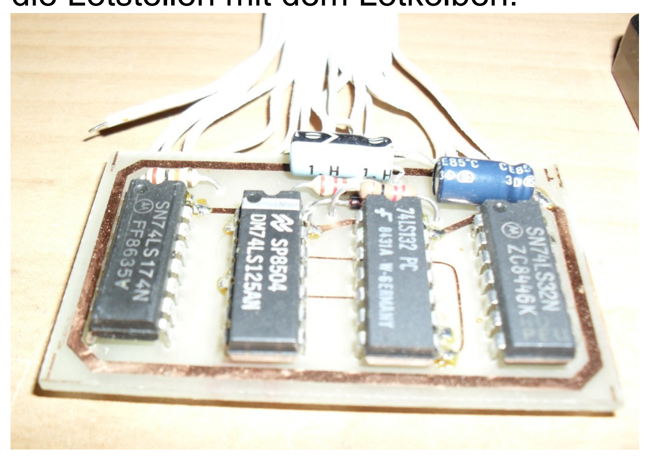
Der Aufrüstsatz besteht aus einem Logikbaustein und acht RAM-Chips des Typs 41256.
Öffnen Sie den SPECTRUM durch Herausdrehen der vier Bodenschrauben, ziehen Sie die Tastaturfolien aus den Buchsen und entfernen Sie das mit dem 5-Volt-Regler verschraubte Kühlblech.

16K-SPECTRUM:

Wenn Sie einen 16K-SPECTRUM besitzen, sehen Sie 12 leere Sockel, die mit IC15 bis IC26 bezeichnet sind. Bestücken Sie sie mit folgenden Bausteinen, wobei unbedingt darauf zu achten ist, dass die Gehäusekerben und die Beschriftungen in die gleiche Richtung wie die der anderen Bausteine zeigen:
IC23: 74LS32 IC24: 74LS00 IC25: 74LS157 IC26: 74LS157
Die restlichen 8 Sockel nehmen die Speicherbausteine 41256 auf.
Alle weiteren Schritte sind bei der 16K- und 48K-Version gleich.

48K-SPECTRUM:

Bei allen 48K-Versionen sind die 32K-DRAM—Chips in die Platine eingelötet und müssen entfernt werden. Wenn Sie eine Vakuum-Entlötstation besitzen, können Sie dies ohne Zerstörung der RAM-Bausteine versuchen. Ein Auslöten mit Hilfe von Lötsauglitze ist fast aussichtslos und fuhrt mit Sicherheit zur Beschädigung der Leiterbahnen. Besser ist die folgende Methode:
Zwicken Sie alle 128 Pins mit einer spitzen Platinenzange oder einer Nagelschere ab. Gut geeignet ist auch eine Miniatur-Bohrmaschine mit eingesetztem Fräser. Entfernen Sie dann die verbliebenen Plastikkörper und spannen Sie die Platine senkrecht stehend in einen Schraubstock. Fassen Sie die Pins von der Bestückungsseite her mit einer Pinzette und erhitzen Sie die Lötstellen mit dem Lötkolben.



Das in den Bohrungen verbliebene Lötzinn können Sie mit Hilfe einer Entlötpumpe oder mit Entlötlitze absaugen. Bevor Sie die Sockel einlöten, sollten Sie Flußmittel- und Lötzinnrückstände mit einem Lösungsmittel (z.B. Aceton) entfernen.
Kontrollieren Sie nun die auf der Bestückungsseite liegenden Leiterbahnen und Lötaugen auf Kurzschlüsse und Unterbrechungen und setzen Sie acht DIL—16-Sockel ein. Verlöten Sie die Pins auf der anderen Platinenseite und überprüfen Sie die

Platine auch hier auf Kurzschlüsse. Biegen Sie Pin 1 aller 41256-Bausteine nach oben und setzen Sie sie in die Sockel (auf richtige Position der Gehäusekerben bzw. Pin-1-Markierungen achten). Verbinden und verlöten Sie die hochgebogenen Pins mit dünnem unisolierten Kupferdraht. Suchen Sie nun die zwischen Pin 9 des IC26 und Pin 9 der 41256-DRAMs verlaufende Leiterbahn auf der Unterseite der Platine und unterbrechen Sie sie an beliebiger Stelle.

ANSCHLUSS DES LOGIKBAUSTEINS:



Auf der Unterseite des Logikmoduls finden Sie 18 Lötpunkte, an die knapp 10 cm lange und beidseitig abisolierte Litzen zu löten sind.

Nehmen Sie dann folgende Verbindungen vor:

Litze	Bez.	Baustein	Pin
1	GND	IC26	8
2	+5V	IC26	16
3	IORQ	IC2	20
4	WR	IC2	22
5	A1	IC2	31
6	RFSH	IC2	28
7	A7	IC15-22	9

Litze	Bez.	Baustein	Pin
8		IC26	9
9	A8	IC15-22	1
10	D1	IC2	15
11	D0	IC2	14
12	D2	IC2	12
13		IC26	10
14	RST	IC2	26
15	STRB	IC26	1

IC1: Z80A-CPU
IC15-22: 41256-DRAMs
IC26: 74LS157/74HC157

Drahtbrücken:
Bei den “Issues" 3 und 4 befinden sich neben der MIC-Buchse einige Drahtbrücken. Setzen Sie die linke obere Brücke ("TI") ein und trennen Sie andere bestehende Verbindungen auf.
Bei der SPECTRUM-Version 2 finden Sie die Brücke zwischen IC1 und IC3. Trennen Sie sie einfach auf.

FUNKTIONSTEST:

Bevor Sie das Gehäuse wieder zusammenbauen, sollten Sie alle Lötstellen noch einmal optisch überprüfen und die Leiterplatte reinigen. Montieren Sie wieder das Kühlblech und schließen Sie Fernseher und Netzteil an. Sollte die Einschaltmeldung nicht erscheinen, ist das Netzteil sofort wieder abzuschalten.
Überprüfen Sie nochmals alle Leitungsverbindungen, die Leiter-bahnen, Lötstellen und die Bauteil—Positionen.
Ist alles in Ordnung, verschrauben Sie

die Platine mit der Bodenschale und suchen eine günstige Stelle für das Logikmodul.
Legen Sie eine Plastikfolie zwischen die Modul-Lötseite und das Kühlblech bzw. die SPECTRUM-Bauteile.
Setzen Sie nun die Platine ins Gehäuse ein, stecken Sie die Tastaturfolien ein und verschrauben Sie das Gehäuseoberteil. Nach dem Anschließen von Fernseher und Netzteil können Sie die Speichererweiterung testen:
PRINT PEEK 23732+256* PEEK 23733
Sie sollten den wert 65535 erhalten.
Setzen Sie dann den RAMTOP mit "CLEAR 32767" herab.
Durch den Umschaltbefehl "OUT 253,n" können Sie nun alle 8 Bänke einschalten, wobei in "n“ die Bank-Nummer (0-7) übergeben wird.
Poken Sie einige Adressen oberhalb von 32768 und lesen Sie sie wieder aus. Wenn Sie vor dem Bank-Wechsel den RAMTOP nicht auf 32767 herabsetzen, stürzt der SPECTRUM ab.

PROGRAMMIER-ANLEITUNG:
Die 256K-Speichererweiterung ersetzt den oberen 32K-RAM-Block des 48K-SPECTRUM durch 8 parallel liegende Bänke zu je 32K. Nach dem Einschalten befindet sich stets Bank 0 im Zugriff der CPU und das Betriebssystem überprüft den "sichtbaren" Speicher, wobei Programm und Daten gelöscht werden. Wenn alle Speicherzellen in Ordnung sind, enthält die Systemvariable P-RAMTOP (23732 und 23733) den Wert 65535, was Sie mit folgender BASIC-Routine feststellen können: PRINT PEEK 23732+256* PEEK 23733
Zwischen P—RAMTOP und RAMTOP (Variable 23730 und 23731) befinden sich der UDG—Zeichensatz und eine Speicherzelle, die die End-Markierung 3E (Hex) enthalt.
Die Maschinen- und GOSUB-Stapel, auf denen die Rücksprungadressen von CALLs und GOSUBs abgelegt sind, schließen sich daran an und dehnen sich je nach Umfang nach unten ins RAM aus.
Es ist nun leicht einzusehen, dass sich die acht 32K—Bänke nicht von BASIC aus umschalten lassen, da dann ja wichtige Informationen verloren gingen und das Programm unweigerlich abstürzen müßte. Ein Ausweg wäre die Begrenzung des BASIC-Teiles auf 16K, was mit CLEAR 32767 leicht erfolgen kann. 48K-Programme wären dann allerdings nicht lauffähig.
Die im folgenden beschriebenen Verfahren greifen über Maschinen= routinen auf die Speicherbänke zu und nutzen sie als "Schatten-RAM". Dabei dient Bank 0 wie bei einem normalen 48K—SPECTRUM als Programm= speicher; die anderen Bänke sind für das Betriebssystem "unsichtbar". Die Vorteile sind eindeutig: Sie können beliebige 48K-Programme ohne Beeinträchtigung fahren und wie gewöhnt in BASIC programmieren.
Die Umschaltung der Bänke erfolgt durch die unteren drei Datenbits, die auf Adresse 253 (=FD) ausgegeben werden. Diese I/O-Adresse ist normalerweise für das RS-232-Interface von SINCLAIR reserviert (Selektion durch A1=LOW). Da dieses nicht sehr verbreitet ist und andere Hardware von A1=HIGH ausgeht, sind keine Komplikationen zu erwarten. Sollten Sie zum exklusiven Kreis der RS232-Interface-Besitzer gehören, können Sie die Bankum= schaltung auf Adresse 223 verlegen:

Verbinden Sie den Anschluß 5 am Logik-Modul nicht mit Pin 31, sondern mit Pin 35 der Z80—CPU. Sie müssen dann allerdings auch die mitgelieferten Maschinenprogramme entsprechend umschreiben. Hardware, die mit A5=LOW in Schreibrichtung angesprochen wird, ist inkompatibel.
Sollten Sie kein Interface-1 benutzen, stehen die Adressleitungen A3 und A4 (Pins 33 und 34 der CPU) zur freien Verfugung. Die I/O-Adressen wären in diesem Fall 247 bzw. 239.

BILDSCHIRM-FILES:
Eine interessante Anwendung ist das Ablegen von hochauflösenden Bildschirmfiles in den sieben Speicherbanken. Jede Bank kann 4 Farb- oder 5 Schwarzweiß-Screens aufnehmen, wobei 5 bzw. 2K je Bank ungenutzt bleiben. Der Bildspeicher ist wie folgt aufgebaut:
16384-22527: 24*32*8 Bildpunkte
22528—23295: 24*32 Attribute (Farbe, Blinken usw.).
Zum Umschalten der Bänke und Verschieben des Bildschirminhaltes dienen zwei kleine Maschinenroutinen. Sie müssen unterhalb der Adresse 32768 abgelegt werden, in unserem Beispiel im Druckerpuffer (23296-23551). Zur Eingabe eignen sich ein Assembler, Monitor oder die im Anhang beschriebene BASIC-Routine. Sollten Sie einen Drucker anschließen, wird der Puffer und damit das Transferprogramm überschrieben. In diesem Fall empfehlen wir die Unterbringung in einem anderen Speicherbereich.

```
ORG 23296
23296 DI             F3 ;Interrupt sperren
23297 LD A,01        3E 01
23299 OUT (253),A D3  FD  ;Einschalten
                          ;von Bank 1
23301 LD HL,16384 21 00 40;Startadresse
                          ;16384
23304 LD DE,32768 11 00 80 ;Zieladresse
                             ;32768
23307 LD BC,6912  01 00 1B ;6912 Bytes
23310 LDIR        ED B0 ;Blocktransfer
23312 XOR A       AF  ;Einschalten von
                          ;Bank 0
23313 OUT (253),A D3 FD
23315 EI          FB ;Interrupt
                          ;freigeben
23316 RET         C9  ;Rücksprung ins
                          ;BASIC
23317 DI          F3 ;Interrupt sperren
23318 LD A,1      3E 01
23320 OUT (253),A D3 FD ;Einschalten
                          ;von Bank 1
23322 LD HL,32768 21 00 80
                  ;Startadresse 32768
23325 LD DE,16384 11 00 40 ;Zieladresse
                             ;16384
23328 LD BC,6912  01 00 1B ;6912 Bytes
23331 LDIR        ED B0 ;Blocktransfer
23333 XOR A       AF
23334 OUT (253),A D3 FD ;Einschalten
                          ;von Bank 0
23336 EI          FB ;Interrupt
                  ;freigeben
23337 RET         C9  ;RUcksprung ins
                          ;BASIC
```

Laden Sie mit LOAD "" SCREEN$ ein Bild und rufen Sie die Verschieberoutine mit RANDOMIZE USR 23296 auf.
Sie können das Bild Jederzeit mit RANDOMIZE USR 23317 wieder in den Bildspeicher zurückholen.
Durch Auswechseln der Werte in den Zellen 23306 und 23324 sind weitere drei Screens in oder aus Bank 1 zu laden. Die Adress-Low-Bytes in den Zellen 23305 und 23323 brauchen nicht geändert zu werden, da die Adressen Vielfache von 256 sind. Die Werte erhalten Sie durch Division der Blockadressen durch 256:
Screen 1: Adresse 32768, Wert 128

(HEX 80)
Screen 2: Adresse 39680, Wert 155 (HEX 9B)
Screen 3: Adresse 46592, Wert 182 (HEX B6)
Screen 4: Adresse 53504, Wert 209 (HEX D1)
Andere Speicherbänke erreichen Sie durch Ändern der Zellen 23298 und 23319.
Mit folgender BASIC-Routine werden in schneller Folge 28 Screens angezeigt:

```
10 FOR b=1 TO 7
20 POKE 23319,b
30 FOR f=0 TO 3
40 POKE 23324,f*27+128
50 RANDOMIZE USR 23317
60 NEXT f
70 NEXT b
```

Natürlich können Sie die Blocktransfer-Routinen auch für andere Zwecke verwenden, wenn Sie die Register BC, DE und HL mit entsprechenden Werten laden. Vergessen Sie jedoch niemals, den Interrupt zu sperren und vor dem Rücksprung ins BASIC wieder freizugeben: Ein Absturz des Programmes bzw. eine Abschaltung der Tastaturabfrage waren die Folgen.

Das nächste MC-Programm dient zum Saven und Laden des gesamten 48K-RAM-Inhaltes, wobei als Speicher=medium kein Kassettenrekorder, Microdrive oder Floppy-Laufwerk dient, sondern die 256K-Erweiterung. Berücksichtigen Sie aber, dass dabei Daten und Programme in den Bänken 1 und 2 überschrieben werden. Als Arbeits-RAM dient stets Bank 0, in die auch nach Abarbeitung der Routinen zurückgesprungen wird.
Das 139 Bytes lange MC-Programm kann im Druckerpuffer, aber auch in jedem anderen Bereich unter 32768 abgelegt werden, wobei sich natürlich die Aufrufadressen entsprechend verschieben.

```
ORG 23296
23296 DI            F3 ;Interrupt sperren
23297 PUSH AF       F5 ;Hauptregister
                           ;retten
23298 PUSH BC       C5
23299 PUSH DE       D5
23300 PUSH HL       E5
23301 PUSH IX       DD E5
23303 PUSH IY       FD E5
23305 LD A,I        ED 57 ;Interrupt
                    ;register retten
23307 LD (23725),A  32 AD 5C
             ;Puffer im Bereich MEMBOT
23310 LD (23726),SP ED 73 AE 5C
                    ;Stapelzeiger retten
23314 LD HL,16384   21 00 40
                    ;Startadresse 16384
23317 LD DE,32768   11 00 80
                    ;Zieladresse 32768
23320 XOR A         AF ;Bank 0
                    ;einschalten
23321 OUT (253),A   D3 FD
23323 LD B,(HL)     46
23324 LD A,01       3E 01
23326 OUT (253),A   D3 FD ;Bank 1
                    ;einschalten
23328 LD A,B        78
23329 LD (DE),A     12
23330 INC DE        13
23331 DEC HL        23
23332 LD A,H        7C
23333 CP 192        FE C0 ;Schleife 1
                    ;beendet?
23335 JR NZ,23320   20 EF ;Wenn nicht,
                    ;wiederholen
23337 LD DE,32768   11 00 80 ;Zieladresse
                    ;32768
23340 XDR A         AF ;Bank 0 einschalten
23341 OUT (253),A   D3 FD
23343 LD B,(HL)     46
23344 LD A,02       3E 02
23346 OUT (253),A   D3 FD ;Bank 2
                    ;einschalten
23348 LD A,B        78
23349 LD (DE),A     12
23350 INC DE        13
23351 INC HL        23
23352 LD A,D        7A
23353 CP 192        FE C0 ;Schleife 2
                    ;beendet?
```

```
23355 JR NZ,23340   20 F1 ;Wenn nicht,
                    ;wiederholen
23357 XOR A         AF
23358 OUT (253),A   D3 FD ;Bank O
                          einschalten
23360 POP IY        FD E1 ;Register wieder
                    ;herstellen
23362 POP IX        DD E1
23364 POP HL        E1
23365 POP DE        D1
23366 POP BC        C1
23367 POP AF        F1
23363 EI            FB ;Interrupt freigeben
23369 RET           C9 ;Rücksprung
```

Das Verschiebeprogramm besteht aus zwei Schleifen, die zuerst 32K in Bank 1 und danach 16K in Bank 2 ablegen. Der komplette RAM-Inhalt einschließlich Bildschirm, Systemvariablen, Stapel und UDG-Zeichensatz wird in die RAM-Erweiterung verschoben.

Zum Rückladen aus der Speichererweiterung dient ein ähnliches Programm:

```
23370 DI            F3 ;Interrupt sperren
23371 LD HL,32768   21 00 80 ;Startadresse
                    ;32768
23374 LD DE,16384   11 00 40 ;Zieladresse
                    ;16384
23377 LD A,01       3E 01 ;Bank 1
                    ;einschalten
23379 OUT (253),A   D3 FD
23381 LD B,(HL)     46
23382 XOR A         AF
23383 DUT (253),A   D3 FD ;Bank O
                    ;einschalten
23385 LD A,B        78
23386 LD (DE),A     12
23387 INC HL        23
23388 DEC DE        13
23389 LD A,D        7A
23390 CP 192        FE C0 :Schleife 1
                    :beendet?
23392 JR NZ,23377   20 EF ;Wenn nicht,
                    ;wiederholen
23394 LD HL,32768   21 00 80 ;Startadresse
                    ;32768
23397 LD A,02       3E 02
23399 DUT (253),A   D3 FD ;Bank 2
                    ;einschalten
23401 LD B,(HL)     46
23402 XDR A         AF
23403 DUT (253),A   D3 FD ;Bank 0
                    ;einschalten
23405 LD A,B        78
23406 LD (DE),A     12
23407 INC HL        23
23408 INC DE        13
23409 LD A,H        7C
23410 CP 192        FE C0 ;Schleife 2
                    ;beendet?
23412 JR NZ,23397   20 EF ;Wenn nicht,
                    ;wiederholen
23414 LD A,(23725)  3A AD 5C
23417 LD I,A        ED 47
              ;Interruptregister laden
23419 LD SP,(23726) ED 7B AE 5C
              ;Stackpointer setzen
23423 POP IY        FD E1 ;Register wieder
                    ;herstellen
23425 POP IX        DD E1
23427 POP HL        E1
23428 POP DE        D1
23429 POP BC        C1
23430 POP AF        F1
23431 IM 1          ED 56;Interrupt-Modus 1
23433 EI            FB ;Interrupt freigeben
23434 RET           C9 ;Rücksprung
```

Nachdem Sie das MC-Programm in den Druckerpuffer eingegeben haben, sollten Sie es vor dem Ausprobieren auf Band aufzeichnen:

SAVE "transfer" CODE 23296,139

Nun können Sie die beiden Routinen testen:

Arbeits-RAM ==> Schatten-RAM:
RANDOMIZE USR 23296

Arbeits-RAM <== Schatten-RAM:
RANDOMIZE USR 23370

wenn Sie ein BASIC-Programm im Speicher haben, das z.B. auf Zeile 10 startet, hängen Sie diese Programmzeile an:

9999 RANDOMIZE USR 23296: GOTO 10

Das Programm wird mit RANDOMIZE USR 23370 wieder in den Arbeitsspeicher zurückgeladen.

Nach dem Ladevorgang wird das Programm mit "GOTO 10" automatisch

gestartet.
Maschinenprogramme lassen sich ebenfalls automatisch aufrufen, wenn Sie anstelle des "GOTO 10“ einen RANDOMIZE USR-Aufruf einsetzen. Beachten Sie bitte, dass sich kommerzielle Spielprogramme meist nicht anhalten und kopieren lassen. Außerdem überschreiben einige Programme den Druckerpuffer, sodass die Transfer-Routine verloren geht.
Wenn Sie ein EPROM-Programmiergerät besitzen, können Sie die 139 Bytes auch ins Betriebssystem des Spectrum einbinden: Dazu ist
der ROM-Inhalt in ein 27128-EPROM zu brennen; die Transferroutine findet z.B. in der ROM-Lücke ab 14446 Platz. Die Aufrufadressen für SAVE und LOAD sind dann 14446 und 14520.
Falls Sie mit dem BETA-Disk-System arbeiten, dürfen Sie die Routine nicht auf Adressen oberhalb von 15350 ablegen, da sonst auf das DOS umgeschaltet wird.
Sie können die Transferroutine auch per Hardware aufrufen: Dazu brauchen Sie nur an den NMI-Eingang der CPU (Pin 17) und GND einen Taster anzuschließen. Ein kurzer Impuls läßt den Prozessor zur Adresse 66 (HEX) im ROM springen. Die dort liegende Routine wird vom SPECTRUM-Betriebssystem nicht benutzt und kann durch eine Sprunganweisung ersetzt werden:

```
00104 JP 14446      C3 5E 3B
```

Auf Tastendruck wird ein laufendes, kopiergeschütztes Programm unterbrochen und eine komplette Kopie in den Banken 1 und 2 abgelegt. Nachdem Sie den RAMTOP mit CLEAR 32767 herabgesetzt haben, können Sie den mitgelieferten Monitor/Disassembler dazuladen und das "geknackte" Programm in Ruhe analysieren.

EXTERNE MASSENSPEICHER:
Der Inhalt der 256K—Speichererweiterung kann auf einfache Weise auf beliebige Massenspeicher übertragen werden.
Zuerst ist der RAMTOP herabzusetzen, dann werden die acht Bänke nacheinander eingeschaltet und auf Band, Microdrive oder Floppy aufgezeichnet:

```
9995 CLEAR 32767
9995 FOR n=0 TO 7
9997 OUT 253,n
9998 SAVE "filename"CDDE 32768, 32768
9999 NEXT n
```

Natürlich können Sie vorher auch einen vorhandenen BASIC-Lader und die im Druckerpuffer befindlichen Maschinenroutinen saven. Das gleiche Verfahren ist auch beim Laden der RAM-Erweiterung anzuwenden. Der zuerst auf Band befindliche BASIC-Teil schaltet hintereinander die einzelnen Bänke ein und lädt sie nach. Zum Schluß kann das BASIC- oder MC-Hauptprogramm aufgerufen werden.
Eines soll aber nicht verschwiegen werden: Nur ein Floppy-Disk-System bietet die Kapazität und Geschwindigkeit, um mit einem derart großen RAM komfortabel arbeiten zu können.

HEX MONITOR:
Das Programm belegt 1170 Bytes im Adreßraum ab 28200 (6E28) und wird auch auf dieser Adresse mit RANDOMIZE USR gestartet.

Alle Ein- und Ausgaben erfolgen sedezimal. Nach dem Befehlskürzel folgen ein oder mehrere Operanten, die mit Leerzeichen zu trennen sind. Führende Nullen sind anzugeben.

Lssss Sedezimal—Listing ab der Adresse ssss. Nach Aufruf des Monitors ist 8000 hex voreingestellt.

C Mit der Taste C wird das Listing fortgesetzt.

D Ausgabe des Listings an den Drucker. Bei eingeschaltetem Drucker blinkt das B in “Befehl".

M Nach Drucken der Taste M verlassen Sie den Monitor und gelangen nach BASIC bzw. ins Menü zurück.

E Der Editor wird eingeschaltet, um Speicherzellen im Anzeigebereich zu andern. Falls sich die gewünschte Adresse nicht auf der Bildschirmseite befindet, ist sie vorher mit der L-Funktion zu listen. Der Cursor wird wie gewohnt mit den Tasten 5 bis 8 gesteuert. Nach jeder Eingabe wird die Adresse erhöht. Durch Drücken der Taste K kommen Sie wieder aus dem Editor heraus.

R Alle Haupt- und Zweit= register der Z80-CPU werden in sedezimaler und binärer Form angezeigt.

Jzzzz Es erfolgt ein unmittelbarer Sprung zur angegebenen Adresse. Falls am Ende des MC-Programmes ein RET (C9) steht, erfolgt eine Rückkehr in den Monitor, wobei alle Register gerettet werden und mit der Funktion R analysierbar sind.

Tssss zzzz bbbb Ein Datenblock der Lange bbbb wird von ssss nach zzzz verschoben. Dabei ist Jedoch darauf zu achten, dass keine Überlappung erfolgt. Um z.B. den gesamten ROM—Inhalt ins RAM zu verschieben, ist folgende Eingabe erforderlich:
T00000 8000 4000
Mit der Transfer-Funktion können Sie auch Speicherbereiche mit Konstanten auffüllen:
Geben Sie mit den L- und E-Funktionen den Wert in die erste aufzufüllende Speicherzelle ein und “vervielfachen“ Sie ihn dann mit dem T-Befehl:
T8000 8001 3FFF
Im Beispiel wird der Bereich von 8000 bis BFFF mit einer auf Adresse 8000 abgelegten Konstanten aufgefullt.

Bsnnnn Der in der Adresse nnnn befindliche Wert wird auf den Stapel gelegt und gegen ein RET ausgetauscht. Ein Maschinenprogramm stoppt an diesem "Breakpoint" und die Register können analysiert werden.

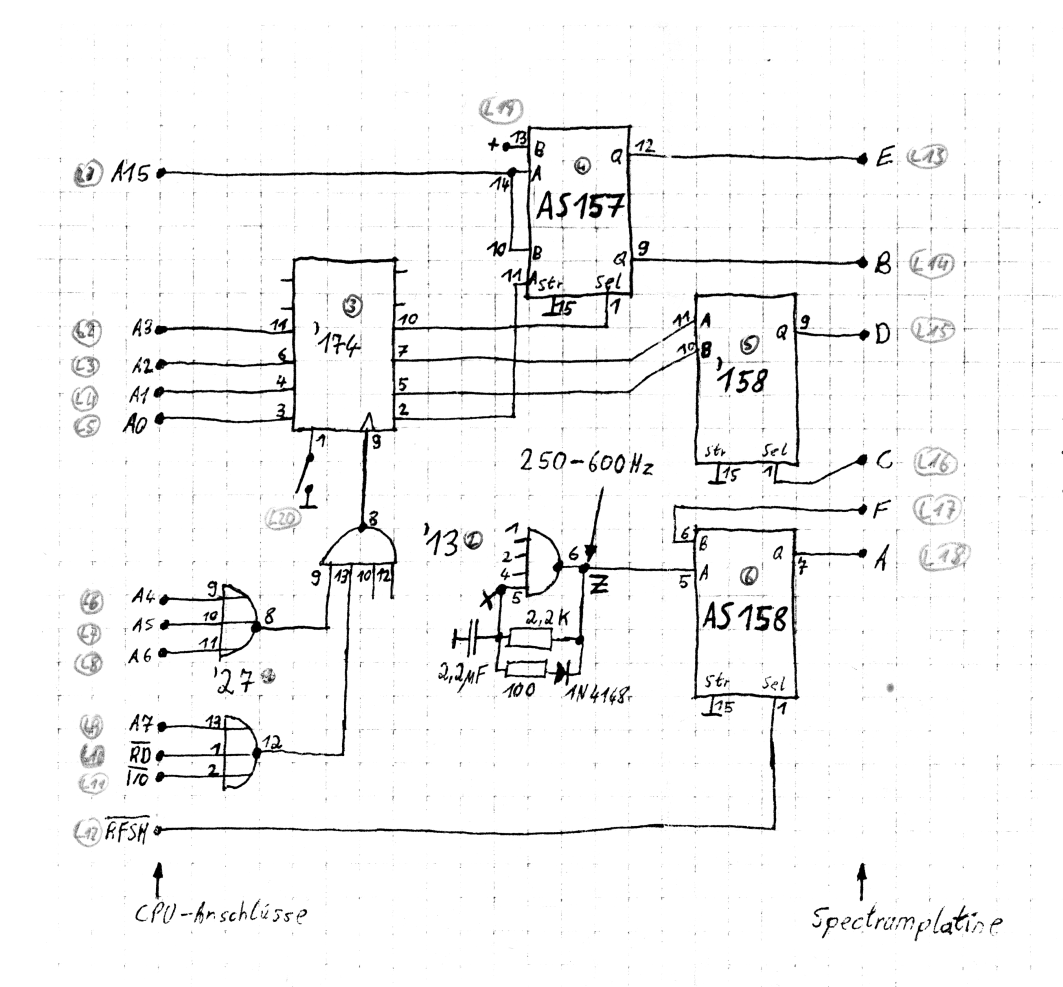
Brnnnn Dieser Befehl holt den alten Wert vom Stapel und setzt ihn in die angegebene Adresse.

Nachwort:

Okay, das war die Einbauanleitung und Nutzungsanleitung für die beiliegende Software Diese wurde eingescannt und OCRt. Auch wenn ich alles überprüft und korrigiert habe, sind Fehler nicht auszuschließen.
Ich werde versuchen die Software auszulesen und eine TZX-Datei daraus zu erstellen und sie auf WOS hochzuladen, allerdings ist die Kassette in keinem hervorragendem Zustand gewesen.

Um nun den Artikel zu vervollständigen, folgt auf nächster Seite die eingescannte Schaltzeichnung.

LCD



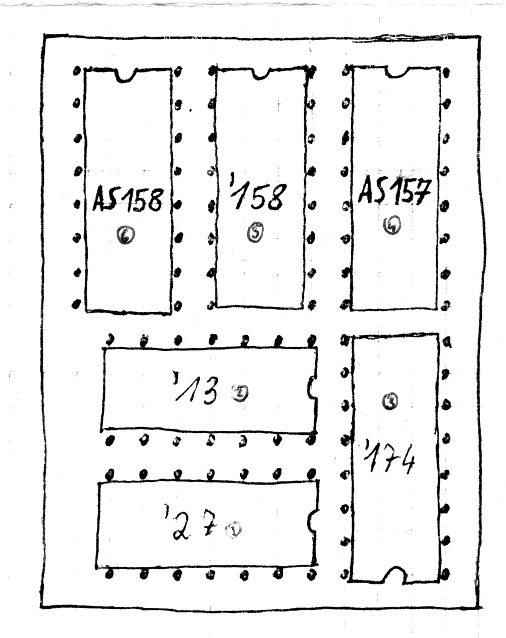
1x 74AS158
1x 74LS174
1x 74AS157
1x 74LS158
1x 74LS13
1x 74LS27
1x 2,2µF
1x 100Ω
1x 2,2kΩ
1x 1N4148

8x 41256

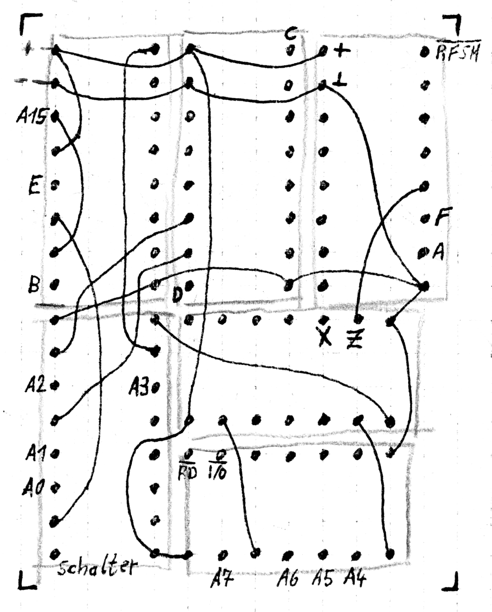
1 Schalter

Lochrasterplatine

Statt „74AS157" und „74AS158" können auch „74S157" und „74S158" verwendet werden.



Oberseite der Platine



Unterseite (Verdrahtung)

X, Z: Bauteile anlöten (siehe Schaltplan)



REPARATUR, ZUBEHÖR & ERSATZTEILE



DEUTSCHLAND

SINTECH.DE LTD
Gastäckerstr. 23
70794 Filderstadt
www.sintech-shop.de



CZECH REPUBLIC

SINTECH.CZ LTD
Masarykova 767
69801 Veseli nad Moravou
www.sintech-shop.cz



UNITED KINGDOM

SINTECH.UK LTD
1 Moorthen Court, Quedgeley
Gloucester, GL2 4LE
www.sintech-shop.co.uk

SINTECH ist ein weltweiter Vertrieb — von Hard- und Software für fast alle Systeme. Sie finden uns in Filderstadt, südlich von Stuttgart.
Desweiteren betreiben wir Niederlassungen in Tschechien und in Großbritannien.

Unser Online-Shop ist mit all unseren Produkten versehen. Immer wieder finden Sie bei uns Neuheiten oder Klassiker in der Rubrik Spectrum Hard- und Software.

Wir schwimmen mal gegen den Strom – mal mit. aber stehen immer für Spectrumfreude pur.

Wann schauen Sie vorbei?



www.sintech-shop.com